

POLITECNICO DI MILANO
Facoltà di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Informatica



Scenari di Rischio e SOA: verso il metodo “Augustus 2.0”

Relatore: Ing. Pierluigi Plebani

Elaborato di Laurea di:

Carlo Carmagnola
Mat. 682772

Carlo Ciminaghi
Matr. 707415

Anno Accademico 2009-2010

Abstract

La gestione dell'emergenza nel panorama italiano nasce dalle intuizioni del Geologo Elvezio Galanti il quale, negli anni 90, ha proposto un approccio organizzativo denominato "Metodo Augustus", universalmente riconosciuto e accettato, il cui punto centrale è la definizione e manutenzione di scenari di rischio.

A quasi venti anni di distanza, il supporto IT a tale metodo risulta essere solo parziale. Alla grande presenza e sviluppo di strumenti hardware e software in dotazione, fa da contraltare la mancanza di standard e di processi volti a garantire l'interoperabilità tra i diversi enti coinvolti.

A livello sintattico gli standard per la codifica di oggetti geo referenziati sono molti e stanno vivendo sviluppi differenti. A livello architetturale, alcuni progetti per la definizione di architetture di interoperabilità hanno gettato le basi per garantire una corretta comunicazione e condivisione delle informazioni.

Con il presente lavoro di tesi, si è analizzato come l'applicazione di metodologie di collaborazione e condivisione tipiche del Web 2.0 possono innovare il supporto informativo al metodo Augustus diventando fattori abilitanti nel percorso che conduce la gestione dell'emergenza verso quello che è proposto come il metodo "Augustus 2.0"

In questo documento, è presentata un'ipotesi di architettura che desidera fornire una soluzione per l'interoperabilità in ambiente EMIS ribadendo il concetto di *Informazione*

come Servizio e soprattutto introducendo il concetto di *Scenario come Servizio* di servizi e di aggiornamento delle informazioni attraverso gli eventi.

Attraverso il concetto di *Informazione come Servizio* si creano i presupposti per la corretta distribuzione della stessa attraverso architetture SOA portando la responsabilità del “discovery” di aggiornamento delle informazioni in rete, dall’utente al fornitore dell’informazione attraverso la creazione e notifica di eventi.

Grazie al concetto di *Scenario come Servizio* si desidera fornire un supporto IT valido alla gestione di scenari di rischio fornendo procedure per la creazione e la gestione di scenari, l’aggiornamento, il salvataggio ed il confronto di dati storici.

Introduzione

La gestione dell'emergenza è uno dei temi maggiormente discussi ed analizzati fin dall'antichità. Il verificarsi di eventi naturali come lo tsunami asiatico del 2004, il terremoto in Abruzzo del 2009 o eventi procurati dall'incuria dell'uomo come lo sversamento di idrocarburi nel fiume Lambro del Marzo 2010, hanno messo in luce come, nonostante gli sforzi, ancora oggi molto si possa compiere nel processo di gestione dell'emergenza, dalla fase di prevenzione al coordinamento delle risorse in campo al verificarsi di un evento.

La gestione dell'emergenza nel panorama italiano trae ispirazione dalle intuizioni del Geologo Galanti il quale, negli anni 90, ha proposto un approccio organizzativo e procedurale denominato "Metodo Augustus". Il metodo Augustus è universalmente riconosciuto e accettato e rappresenta la base della Protezione Civile Italiana.

A quasi venti anni di distanza, il supporto IT a tale metodo risulta essere solo parziale. Alla grande presenza e sviluppo di strumenti hardware e software in dotazione a operatori e sale di controllo, fa da contraltare la mancanza di standard e di processi volti a garantire l'interoperabilità tra sale di controllo ed enti facenti parte del processo di gestione dell'emergenza. Punto centrale del metodo Augustus è la definizione di uno scenario di rischio, come raggruppamento di informazioni geolocalizzate, sensori, strutture, risorse.

A livello sintattico gli standard (o proposte in itinere) per la codifica di oggetti geo referenziati e reti di sensori basate sul meta-linguaggio XML sono molti e stanno vivendo sviluppi differenti. A livello architetturale, alcuni progetti per la definizione di architetture di interoperabilità hanno gettato le basi per garantire una corretta

comunicazione e condivisione delle informazioni in modo trasparente alle diverse tecniche implementative o di comunicazione.

Con il presente lavoro di tesi, si è analizzato come l'applicazione di metodologie di collaborazione e condivisione tipiche del Web 2.0 possono innovare il supporto informativo al metodo Augustus che al momento trova nell'ICT diverse soluzioni scarsamente integrabili. Il supporto stesso diventa quindi fattore abilitante nel percorso che conduce la gestione dell'emergenza verso quello che è proposto come il metodo "Augustus 2.0"

Inoltre, in questo documento, è presentata un'ipotesi di architettura che basandosi su quanto già esistente, soprattutto sul progetto Europeo Orchestra, desidera fornire una soluzione per l'interoperabilità in ambiente EMIS ribadendo il concetto di *Informazione come Servizio* e soprattutto introducendo il concetto di *Scenario come Servizio* di servizi e di aggiornamento delle informazioni attraverso gli eventi.

Attraverso il concetto di *Informazione come Servizio* si creano i presupposti per la corretta distribuzione della stessa attraverso architetture SOA garantendo altresì la correttezza dei dati grazie allo spostamento della responsabilità del "discovery" di aggiornamento delle informazioni in rete, dall'utilizzatore al fornitore dell'informazione attraverso la creazione e notifica di eventi.

Grazie al concetto di *Scenario come Servizio* si desidera fornire un supporto IT valido alla gestione di scenari di rischio fornendo procedure per la creazione e la gestione di scenari, l'aggiornamento, il salvataggio ed il confronto di dati storici, estendendo l'architettura Orchestra.

Segue una breve presentazione dei diversi capitoli che compongono il lavoro di tesi.

Nel Capitolo 1 è presentato il dominio applicativo all'interno del quale si intende calare le soluzioni ed i concetti successivamente proposti: il processo di gestione dell'emergenza, il metodo Augustus base della Protezione Civile Italiana, l'importanza dei piani di emergenza e della gestione delle risorse.

Nel Capitolo 2 sono presentati i principali strumenti utilizzati ad oggi nella gestione delle emergenze: i GIS, le reti di sensori e gli strumenti a disposizione delle Sale di Controllo evidenziando alcune criticità, con particolare enfasi alla mancanza di interoperabilità, alle quali si desidera dare risposta con questo documento.

Nel Capitolo 3 sono descritti i principali standard ad oggi esistenti o in via di definizione che contribuiscono a creare i presupposti per la definizione di sistemi interoperabili: dagli standard più generici (Web Service e Architetture SOA) a i progetti più legati al dominio degli EMIS presentando inoltre alcune soluzioni già ipotizzate.

Nel Capitolo 4 è presentato un nuovo approccio al metodo Augustus, basato sulla filosofia 2.0 attraverso la definizione dei requisiti di dominio e dell'architettura attraverso la quale supportare Augustus: 2.0: informazione come servizio, l'aggiornamento delle informazioni tramite eventi, il concetto di scenario come servizio di servizi e le sue possibili estensioni.

Nel Capitolo 5 è descritta una possibile definizione architeturale dei concetti espressi al Capitolo 4 quali la gestione di eventi in Risk Management Information System e il concetto di Scenario come servizio di servizi sfruttando l'architettura Orchestra.

Nel Capitolo 6 sono presentati due scenari al fine di validare le peculiarità dell'architettura "sul campo". Gli scenari utilizzati sono il rischio di esondazione del fiume Lambro e il rischio di una frana a Premana in Provincia di Lecco.

Sommario

Abstract	1
Introduzione.....	3
Sommario.....	6
Indice Figure.....	13
Capitolo 1 Il dominio Applicativo: Gestire l'emergenza.....	16
1.1 Gestione dell'emergenza	16
1.1.1 Preparedness.....	17
1.1.2 Response	18
1.1.3 Recovery.....	19
1.1.4 Mitigation	19
1.2 Metodo Augustus.....	20
1.3 Piano di Emergenza.....	23
1.3.1 Scenario di Rischio.....	23
1.3.2 Aggiornamento Periodico	24
1.3.3 Attuazione di Esercitazioni.....	25
1.3.4 Informazione alla Popolazione.....	25
1.4 Gestione delle Risorse.....	26
1.4.1 La disponibilità delle Risorse	26
1.4.2 La Pre-Allocazione di Risorse.....	27

Capitolo 2 Tecnologie IT a supporto delle Attività di Gestione delle Emergenze 28

2.1 GIS: Geographic Information Sytem	28
2.1.1 Ambiti d'uso e funzionalità di un GIS	29
2.1.2 Tecnologie e Tecniche GIS.....	30
2.1.3 Gli sviluppi del GIS, gli standard e il web	31
Open Geospatial Consortium	31
2.2 Sensori e Sistemi di Monitoraggio	32
2.3 Sale di Controllo.....	33
2.3.1 Sistema Informativi ed integrazione dei Dati.....	34
Integrazione di Dati e Sensori.....	34
Integrazione tra le diverse funzioni all'interno della Sala	34
Gestione del flusso delle informazioni in entrata ed in uscita.....	34
Integrazione tra Sale di Controllo.....	35

Capitolo 3 Interoperabilità negli Emergency Management Information System 36

3.1 Gli standard operativi	36
3.2 Servizi Web e Catalogazione	37
3.2.1 Web Service.....	38
3.2.2 Servizi Catalogo OGC Catalog e UDDI.....	38
3.3 Geolocalizzazione e Sensori	39
3.3.1 WMS - Web Map Service	39
3.3.2 WFS – Web Feature Service	40
3.3.3 SWE – Sensor Web Enablement.....	42

Observation & Measurements Schema (O&M)	43
Sensor Model Language (SensorML):.....	44
Transducer Markup Language (TML).....	44
Sensor Observation Service (SOS)	44
Sensor Planning Service (SPS).....	44
Sensor Alert Service (SAS)	44
Web Notification Services (WNS)	45
3.3.4 CAP – Common Alerting Protocol	45
3.3.5 Un esempio di Implementazione: FrameWork MapServer	45
3.4 Interoperabilità di EMIS	46
3.4.1 Progetto Orchestra.....	47
3.4.2 OASIS Fp6	49

Capitolo 4 Augustus 2.0: Informazione e Scenario in ambiente SOA ... 52

4.1 Web 2.0 Collaborare e Condividere.....	52
4.1.1 SOA – La collaborazione attraverso i servizi	53
4.1.2 Il web 2.0 in scenari d'emergenza.....	54
Terremoto in Abruzzo.....	54
Terremoto Haiti	55
4.2 Verso Augustus 2.0	55
4.2.1 Collaborazione.....	56
4.2.2 Conoscenza Distribuita e Condivisa	56
4.2.3 Flessibilità	57
4.3 Architettura: Requisiti Architettonici	58
4.3.1 L'informazione come Servizio	59

Architettura per la Condivisione delle Informazioni	59
Aggiornare le Informazioni tramite Eventi	61
Ricerca le informazioni in Ambiente Distribuito	64
4.3.2 Lo Scenario come Servizio di Servizi.....	68
Il servizio scenario	68
Database Temporale	70
Evoluzione basata su dati Storici	70
Lo scenario all'interno di Augustus 2.0	71
Capitolo 5 Principi Architettureali	73
5.1 L'architettura Orchestra	73
5.1.1 Domini Funzionali dell'ORCHESTRA Service Network.....	74
5.1.2 L'interoperabilità semantica: Information Meta-model e Service Meta-model	76
Information Meta-model.....	77
Service Meta-model	78
5.1.3 I Nodi Orchestra	79
5.1.4 Il servizio Base Orchestra: OA Basic Service.....	80
5.1.5 Authentication e Authorization.....	82
Authentication Service	83
Authorization Service	84
5.1.6 Feature Access Service	85
5.1.7 Catalogue Service	87
5.1.8 Map and Diagram Service	89
5.2 Estensione: Gestione Eventi	89

Observer Pattern	90
Estensione dell' OA Basic Service	91
Interfaccia Service Event	92
5.3 Estensione: Concetto Scenario	94
5.3.1 Nuovi OA_Types	95
OA_ScenarioID.....	95
OA_ScenarioDescriptor	96
OA_ScenarioData.....	96
Ipotesi implementativa di OA_ScenarioData	96
OA_ScenarioConfiguration	97
Ipotesi implementativa di OA_ScenarioConfiguration.....	97
OA_ScenarioObject.....	98
5.3.2 Eccezioni	98
5.3.3 Interfaccia ServiceCapabilities	99
getCapabilities	99
5.3.4 Interfaccia ScenarioAdministration.....	99
addScenarioDescription	100
removeScenarioDescription	100
updateScenarioDescription	100
setScenarioData.....	100
5.3.5 Interfaccia ScenarioConfiguration.....	102
setScenarioConfiguration	102
getScenarioConfiguration.....	103
5.3.6 Interfaccia ScenarioData	103

getScenarioData	103
getScenarioHistory	104
5.3.6 Interfaccia ScenarioEvent.....	104
addScenarioEventListener	104
updateScenarioEventListener	105
removeScenarioEventListener.....	105
Capitolo 6 Scenari di Validazione	106
6.1 La Validazione dell'architettura	106
6.2 Il caso Lambro	107
6.2.1 Creazione dello Scenario: caso pratico	107
Login e inizio creazione dello scenario	108
Ricerca della mappa della zona interessata	109
Ricerca layer aggiuntivi: altimetria e composizione del terreno.....	110
Utilizzo del servizio di simulazione: valutazione delle zone a rischio	111
Visione unificata della mappa	112
Ricerca dei sensori idrometrici	113
Ricerca di altri sensori a monte dell'area di interesse	114
Attivazione del servizio di notifica.....	115
Ricerca di sensori meteorologici	117
Attivazione del servizio di notifica dei sensori	117
Ricerca Strutture a rischio ambientale	118
Attivazione del Servizio di Auto Aggiornamento dello Scenario	119
Ricerca Strutture Pronto intervento.....	121
Ricerca documenti report dell'area interessata	122

Aggiunta documento di analisi del rischio	123
Pubblicazione dello scenario	123
Inserimento dati aggiornati nello scenario	124
6.2.2 Valutazione della situazione Attuale di uno Scenario.....	125
6.2.3 Ricerca di Dati Storici	127
Ricerca di scenari simili.....	127
Recupero scenari passati	129
6.3 Il caso Premana	129
6.3.1 Notifica di Evento su scenario	130
Ricerca Scenario di Premana	131
Lettura configurazione e analisi dello scenario	131
Attivazione servizio notifica eventi	133
Conclusioni.....	134
Bibliografia.....	135

Indice Figure

Figura 1 - Le fasi di gestione dell'emergenza	17
Figura 2 - Sensor Web (OGC White Paper – OGC Sensor Web Enablement: Overview And High Level Architecture.).....	33
Figura 3 - Esempio di TSO	51
Figura 4 - Forum Corriere della Sera - Terremoto in Centro Italia	55
Figura 5 - Architettura e Linguaggi. Peculiarità	61
Figura 6 - Topologia Rete Informazione Utilizzatore.....	63
Figura 7 - Ricerca in cascata in ambiente distribuito	67
Figura 8 - Esempio di Scenario.....	69
Figura 9 - Utilizzo del Database Temporale.....	70
Figura 10 - Domini Funzionali in un Orchestra Service Network (RM-OA Rev2.1 as OGC Best-practice document 07-097).....	76
Figura 11 - Parte principale del Orchestra Information Meta-model (RM-OA Rev2.1 as OGC Best-practice document 07-097).....	78
Figura 12 - Modelli di Informazione di Orchestra (RM-OA Rev2.1 as OGC Best-practice document 07-097).....	79
Figura 13 - Interfaccia Service Capabilities (RM-OA Rev2.1 as OGC Best-practice document 07-097).....	81
Figura 14 - Interfaccia Synchronous Interaction (RM-OA Rev2.1 as OGC Best-practice document 07-097).....	81
Figura 15 - Interfaccia Asynchronous Interaction (RM-OA Rev2.1 as OGC Best-practice document 07-097).....	82
Figura 16 - Interfaccia del Servizio Feature (RM-OA Rev2.1 as OGC Best-practice document 07-097).....	86

Figura 17 - Diagramma UML dell' observer pattern.....	90
Figura 18 - Class Diagram dell' Oggetto OA_Notification.....	91
Figura 19 - Diagramma UML dell'interfaccia Event Service	92
Figura 20 - OA Types dell'interfaccia Service Event	93
Figura 21 - Scenario come contenitore di oggetti (Servizio di Servizi).....	95
Figura 22 - OA_ScenarioData - dataType	96
Figura 23 - setScenarioData come OA_EventOperation	101
Figura 24 - Registrazione e notifica di Evento tra Scenario e Informazione Generica .	102
Figura 25- Login e inizio composizione scenario	109
Figura 26- Ricerca Mappa	110
Figura 27- Aggiunta layer tematici alla mappa.....	111
Figura 28- Servizio Simulazione e Analisi del rischio	112
Figura 29 - Visione unificata della mappa	113
Figura 30 - Ricerca sensori idrometrici	113
Figura 31 - Ricerca sensori idrometrici	114
Figura 32 - Attivazione evento.....	115
Figura 33 - Notifica dell'evento	116
Figura 34 - Attivazione del servizio di notifica dei sensori	118
Figura 35 - Ricerca Strutture a rischio ambientale.....	119
Figura 36 - Attivazione del Servizio di Auto Aggiornamento.....	120
Figura 37 - Auto Aggiornamento dello scenario.....	121
Figura 38 - Ricerca documenti report dell'area interessata.....	122
Figura 39 - Aggiunta documento di analisi del rischio	123
Figura 40 - Pubblicazione dello scenario	124
Figura 41 - Inserimento dati aggiornati nello scenario	125
Figura 42 - Valutazione della situazione attuale dello scenario.....	126
Figura 43 - Sottoscrizione del servizio aggiornamento	127
Figura 44 - Ricerca di scenari simili.....	128
Figura 45 - Recupero scenari passati.....	129
Figura 46 - Scenario di Rischio Cartaceo Premana	130

Figura 47 - Ricerca Scenario di Premana	131
Figura 48 - Recupero dei dati a partire dalla configurazione di Scenario	132
Figura 49 - Lettura configurazione e analisi dello scenario	132
Figura 50 - Attivazione servizio notifica eventi	133

Capitolo 1

Il dominio Applicativo: Gestire l'emergenza

Scopo del capitolo è presentare il dominio applicativo all'interno del quale si intende calare le soluzioni ed i concetti successivamente proposti. La gestione dell'emergenza è un processo complesso ed articolato che richiede grandi sforzi di analisi e organizzazione del territorio attraverso strumenti condivisi e capaci di adattarsi ad eventi di diversa natura ed entità.

1.1 Gestione dell'emergenza

La gestione dell'emergenza è uno dei temi maggiormente discussi ed analizzati fin dall'antichità. Il verificarsi di eventi naturali come lo tsunami asiatico del 2004, il terremoto in Abruzzo del 2009 o eventi procurati dall'incuria dell'uomo come lo sversamento di idrocarburi nel fiume Lambro del Marzo 2010, hanno messo in luce come, nonostante gli sforzi, ancora oggi molto si possa compiere nel processo di gestione dell'emergenza, dalla fase di prevenzione al coordinamento delle risorse in campo al verificarsi di un evento.

La gestione dell'emergenza, come le altre discipline scientifiche, ha giovato delle migliorie tecnologiche e organizzative realizzate nel corso degli anni. A livello italiano,

esperienze come il terremoto in Friuli e soprattutto in Irpinia hanno dato impulso alla creazione di un ente preposto alla gestione delle emergenze, la *Protezione Civile* la quale ha come mandato l'organizzazione e l'applicazione delle migliori tecnologie a disposizione al fine di ridurre al minimo le conseguenze sulla normale vita umana di eventi "calamitosi".

Nell'ambito della ricerca scientifica applicata alla gestione dell'emergenza grande rilievo ha conquistato l'analisi della gestione dell'emergenza vista come processo di collaborazione tra enti e soggetti secondo le diverse funzioni e peculiarità.

Nella rappresentazione classica, ormai ampiamente diffusa e riconosciuta, la gestione dell'emergenza risulta un processo ciclico all'interno del quale, nelle diverse fasi, si indicano le azioni da intraprendere al fine di prepararsi all'evento, ridurre gli effetti al minimo possibile e riportare il più velocemente possibile la situazione alla normalità. All'interno del processo di gestione dell'emergenza si individuano 4 fasi principali (Figura 1) che descrivono un processo ciclico e continuativo scandito da alcuni eventi che determinano, in modo più o meno netto, il passaggio da una fase alla successiva.



Figura 1 - Le fasi di gestione dell'emergenza

1.1.1 Preparedness

Obiettivo della fase di preparedness è la creazione di un adeguato livello di prontezza nel rispondere alle emergenze che nel territorio possono verificarsi. Tale preparazione

si deve ottenere ai diversi livelli organizzativi e decisionali (governo, regioni, comunità, individui) attraverso analisi dei rischi, circolazione delle informazioni e creazione di piani di risposta condivisi.

In questa fase è fondamentale un corretto processo di analisi dei rischi svolto dalle autorità competenti al fine di valutare la possibilità che un evento si verifichi, l'intensità prevista, l'interconnessione con altri eventi calamitosi.

Alla conoscenza dei rischi potenziali deve seguire una fase di diffusione della conoscenza creata nella comunità al fine di fornire i primi e più semplici strumenti di autodifesa al verificarsi dell'evento (punti di raccolta, comportamenti da seguire, comportamenti da evitare...). In questa fase vengono attivati i processi di creazione di scenari di rischio, piani di emergenza, censimento delle risorse procedure di attivazione delle risorse.

1.1.2 Response

Il verificarsi dell'evento calamitoso determina l'inizio della fase di *Response*. La fase di risposta all'emergenza è la fase operativa in cui si mettono in pratica le procedure e i processi descritti nei piani di emergenza. All'interno di questa fase trovano spazio tutte le attività legate all'attivazione delle risorse disponibili per la gestione dell'emergenza in atto, dalla ricerca di feriti e superstiti alla creazione di rifugi o luoghi di ricovero, dalla creazione dei diversi centri operativi di coordinamento e controllo, al passaggio a regimi di legislazione e regolamentazione propri delle emergenze.

È in questa fase che in più di ogni altra si misura e si beneficia del lavoro svolto in fase di preparazione, della definizione di piani di emergenza, dell'istituzione di processi e programmi operativi. L'efficacia stessa della fase di risposta dipende da molte variabili, prime fra tutte il tempo di intervento e la disponibilità di informazioni corrette ed aggiornate, a supporto dei processi di Decision Making.

La fase di risposta risente inoltre di tutte le altre variabili legate agli aspetti fisici e naturali dell'evento che, come intuibile, differiranno da quanto previsto in sede di

definizione del Piano di Emergenza e che dovranno essere gestite in real-time senza il supporto di una analisi preventiva approfondita svolta “in tempo di pace”. E’ il momento in cui occorre prendere decisioni giuste e tempestive ricorrendo il più possibile a quanto progettato e strutturato.

La fase di risposta è anche una fase di verifica in cui si può valutare il lavoro svolto nella fase precedente (o in caso di eventi calamitosi ciclici, nelle fasi precedenti) allo scopo di migliorare nei punti di maggiore criticità la gestione dell’emergenza stessa per il futuro.

1.1.3 Recovery

La fase di recovery si pone come obiettivo il ritorno alla normalità nel più breve tempo possibile. Si attiva la fase di recovery non appena si considera chiusa l’emergenza in senso stretto, ovvero nel momento in cui si è provveduto a mettere in sicurezza persone e patrimonio colpito dall’evento. Nella fase di recovery l’attenzione è posta tutta nel ripristino della normalità, da un punto di vista economico/produttivo, sociale e urbanistico/paesaggistico.

E’ il momento chiamato comunemente “ricostruzione”, da svolgere con particolare attenzione, al fine di gettare le basi per evitare in futuro l’emergenza o per ridurne i suoi effetti dannosi su persone e cose.

1.1.4 Mitigation

L’ultima fase del processo è la fase di mitigation. E’ una fase di lungo periodo in cui ricercare le contromisure a tutti i livelli per evitare il ripetersi dell’evento “calamitoso” o per ridurne le conseguenze, in accordo e continuazione con la fase di recovery. Tale operazione si può svolgere con diverse modalità a seconda del pericolo e del rischio presente nell’area. Si può cercare di agire sul pericolo stesso cercando di eliminarlo o ridurlo (es. rinforzare gli argini di un fiume, realizzare costruzioni antisismiche) oppure si può cercare modificare la modalità con cui l’evento interagisce con la popolazione

(es. creando sistemi di allerta in tempo reale, come nel caso dello Tsunami del sud-est asiatico) oppure tentare, attraverso campagne di informazione e di comunicazione, di modificare gli stili di vita delle persone, le loro abitudini ed i loro comportamenti al fine di renderli edotti del rischio e di poter quindi prendere precauzioni di primo intervento (es. Creazione di Punti di raccolta o esercitazioni, cartellonistica)

La fase di Mitigazione deve sfociare in una fase di preparazione, re iniziando così il ciclo, allo scopo, qualora non fosse possibile evitare il rischio, di migliorare la risposta e ridurre gli effetti negativi.

1.2 Metodo Augustus

Il “Metodo Augustus” (Galanti, 1997) nasce dalle intuizioni del Geologo Elvezio Galanti che alla luce delle disposizioni contenute nella legge 255/92 propone una metodologia di pianificazione dell'emergenza basata sui concetti di flessibilità e semplicità.

Il nome Augustus deriva dall'affermazione dell'imperatore romano Augusto secondo il quale “il valore della pianificazione diminuisce con la complessità dello stato delle cose” e che già allora metteva in evidenza come il concetto di complessità fosse intrinsecamente contrario alla pianificazione preventiva.

Conscio della forte complessità che un “evento d'emergenza” custodisce al proprio interno, per la complessità stessa delle leggi della fisica e della natura che l'evento coinvolge, Galanti sviluppa una nuova metodologia integrata che possa risultare integrata con l'organizzazione e la struttura socio-politica della realtà italiana. Al concetto più classico del “divide et impera” il metodo Augustus aggiunge alcune linee guida per la gestione ed il coordinamento delle diverse attività principali e di supporto alla gestione dell'emergenza. Il tema del coordinamento delle forze in campo è infatti uno dei temi centrali più volte sottolineato da Galanti, il quale evidenzia come nella neonata Protezione Civile esistano *“una miriade di proposte spesso in contraddizione tra loro perché formulate dalle varie amministrazioni locali e centrali”* o come *“nel*

nostro paese non mancano i materiali ed i mezzi: mancano soprattutto gli indirizzi sul come attivare queste risorse in modo sinergico”.

Nel dettaglio Augustus intende essere un metodo flessibile e non legato al dominio di un particolare evento o rischio, un metodo semplificato nell'individuazione delle competenze e del “chi fa cosa”; Augustus introduce il concetto di disponibilità delle risorse, di “piano di emergenza” vivo e pronto al cambiamento, di funzioni di supporto e di gestione dell'incertezza.

Il metodo Augustus identifica e concentra le funzioni di coordinamento dei diversi livelli amministrativi (Comuni, Province, Regioni, Stato Centrale) presso i Centri Operativi definendo:

- C.O.C.: Centro operativo comunale, presieduto dal sindaco o da altro soggetto da esso delegato
- C.O.M: Centro Operativo Misto, strutture a livello provinciale o sovracomunale con lo scopo di coordinare al meglio centri comunali e sindaci afferenti al COM stesso.
- C.C.S: Centro di Coordinamento Soccorsi, struttura provinciale presieduta dal Prefetto o suo delegato che funge da ente per il coordinamento delle forze in campo per la gestione dell'emergenza. Il CCS è inoltre deputato alla gestione delle richieste di soccorso raccolte attraverso la Sala Operativa Provinciale.
- C.O.R. Comitato Operativo Regionale presieduto dal presidente della regione
- Direzione di comando e Controllo Nazionale con sede presso il Dipartimento della Protezione civile

Secondo la metodologia Augustus i diversi centri operativi debbono mettere a disposizione 14 funzioni di supporto indipendenti dal dominio dell'emergenza al fine di coordinare ed erogare risposte operative. Tali funzioni risultano ridotte a 9 nel caso di Centri Operativi Comunali. Le 14 funzioni individuate nel lavoro di Galanti sono:

- **F 1 - Tecnica e di pianificazione:** Gruppi di ricerca scientifica (CNR) - Istituto Nazionale di Geofisica - Regioni - Dipartimento PC - Servizi Tecnici Nazionali

- **F 2 - Sanità, assistenza sociale e veterinaria:** Responsabile C.O. 118 - Regione / AA.SS.LL - C.R.I. - Volontariato socio-sanitario
- **F 3 - Mass-media e informazione:** RAI - Emittenti TV / radio private: nazionali e locali - Stampa
- **F 4 – Volontariato:** Dipartimento PC - Associazioni locali, provinciali, regionali, nazionali
- **F 5 - Materiali e mezzi:** C.A.P.I. - Ministero dell'Interno - Sist. Mercurio . FF.AA - C.R.I. - Aziende pubbliche e private - Volontariato
- **F 6 - Trasporti, circolazione e viabilità:** FF.SS. - Trasporto gommato, marittimo, aereo - ANAS - Soc. Autostrade - Province - Comuni - ACI
- **F 7 – Telecomunicazioni:** Ente Poste - Ministero delle Telecomunicazioni - Acconto di Telecomunicazioni
- **F 8 - Servizi essenziali:** ENEL - ENI - Gas - Acquedotto - Aziende Municipalizzate - Sistema bancario - Distribuzione carburante - Attività scolastica
- **F 9 - Censimento danni a persone e cose:** Attività produttive (ind., art., comm.) - Opere pubbliche - Beni culturali - Infrastrutture - Privati
- **F 10 - Strutture operative:** Dipartimento PC - VV.F. - Forze Armate - C.R.I. - C.C. - G.D.F. - Forestale - Capitanerie di Porto - P.S. - Volontariato - CNSA (CAI)
- **F 11 - Enti locali:** Regioni - Province - Comuni - Comunità montane
- **F 12 - Materiali pericolosi:** VV.F. - C.N.R. - Depositi e industrie a rischio
- **F 13 - Assistenza alla popolazione:** Forze Armate - Ministero dell'Interno - C.R.I. - Volontariato - Regioni - Province - Comuni
- **F 14 - Coordinamento centri operativi:** Collegamento con i centri operativi misti - Gestione delle risorse - Informatica

A capo di ogni funzione di supporto vi è un responsabile cui spetta il compito in “tempo di pace” di mantenere, aggiornare e tenere vive le dinamiche di attivazione e gestione della singola funzione.

Lo schema logico dell’organizzazione che ne deriva è uno schema di tipo piramidale nel quale ad ogni livello superiore spetta il coordinamento e la gestione dei livelli inferiori,

pur non mancando una decisa autonomia dei diversi livelli i quali, all'interno di procedure e di suddivisione delle competenze e delle responsabilità definite a priori, risultano in grado di operare con efficacia sul territorio. Risulta altresì evidente come la gestione di una struttura di coordinamento così flessibile e semplice nella sua articolazione richieda un elevato flusso di informazioni ed una continua operatività al fine di tenere vivi in "tempo di pace" quei meccanismi la cui efficienza ed efficacia risultano fondamentali allo scatenarsi dell'evento.

1.3 Piano di Emergenza

Il piano d'emergenza identificato dalla metodologia Augustus è un documento (o insieme di documenti) rappresentante il supporto operativo al quale le autorità si riferiscono per la gestione dell'emergenza perseguendo la massima efficienza possibile. Il piano di emergenza è il risultato di uno studio approfondito sulla vulnerabilità del territorio, sul rischio o possibilità che questo sia investito da un evento calamitoso e sui punti critici che all'interno del territorio stesso possono risentire con diversa sensibilità del verificarsi dell'evento. Il Piano assume quindi la forma di un progetto che, a partire da alcuni eventi o situazioni contingenti, descrive lo stato e il work flow della attività da svolgere al fine di garantire un rapido intervento delle risorse presenti e necessarie ed un conseguente ritorno alle normali condizioni di vita.

1.3.1 Scenario di Rischio

Alla base della costruzione di un Piano di Emergenza, conseguenza dello studio del territorio sopra descritto, vi è la definizione di uno o più scenari di rischio sulla base della vulnerabilità del territorio interessato. All'interno dello scenario trova spazio una mappatura del territorio stesso nei termini di strutture produttive, popolazione, ambienti naturali, urbani ed industriali, viabilità, sistemi di distribuzione di acqua, luce e gas e tutte le altre entità che potrebbero essere danneggiate dal verificarsi di uno o

più eventi calamitosi. Lo scenario di rischio descrive quindi un quadro globale e verosimile all'interno del quale occorre definire le contromisure o la necessaria risposta operativa in termini di risorse (mezzi dei vigili del fuoco, volontari, strutture di controllo...) o di procedure (vie di fuga, raccolta della popolazione e messa in sicurezza...)

Risulta chiaro, come confermato dallo stesso Galanti nella definizione del metodo Augustus, che il piano di emergenza non può essere un documento che una volta redatto *“resta nel fondo di un cassetto”* in quanto per mantenere la propria globalità e verosimiglianza occorre che il Piano sia la rappresentazione che più si avvicina alla realtà di un territorio. Lo scenario di rischio stesso deve essere continuamente aggiornato e ricontrollato evitando così che informazioni errate o semplicemente non aggiornate intacchino l'efficacia del piano stesso.

A supporto del processo di realizzazione di un Piano di Emergenza, il metodo Augustus definisce tre attività da svolgere in modo continuativo durante tutto il ciclo di vita del Piano: Aggiornamento Periodico, Attuazione di Esercitazioni e Informazione alla popolazione.

1.3.2 Aggiornamento Periodico

La necessità di un aggiornamento periodico risulta evidente in quanto il Piano e lo Scenario di Rischio risentono fortemente della *“dinamicità dell'assetto del territorio”*. L'apertura di un nuovo centro commerciale, la costruzione di una nuova linea metropolitana o di un nuovo quartiere, la semplice modifica della viabilità ordinaria o l'aumento o la diminuzione del numero di studenti medio di una scuola sono dati che possono inficiare l'efficacia di un Piano di Emergenza. A questo occorre aggiungere la disponibilità dei mezzi di soccorso e delle risorse in genere, oltre all'aggiornamento delle tecnologie di controllo ed alla ricerca scientifica che nel tempo può portare a nuove e diverse soluzioni per la gestione dell'emergenza.

Il metodo Augustus individua quindi la necessità di un aggiornamento periodico delle informazioni, secondo le varie responsabilità e competenze con lo scopo di avere, al verificarsi dell'evento, un documento il più verosimile possibile, dalle piccole modifiche all'intera messa in discussione dell'efficacia del piano.

1.3.3 Attuazione di Esercitazioni

Lo svolgersi di esercitazioni basate su uno scenario di emergenza ha il duplice scopo di verificare la validità del piano e dei suoi eventuali aggiornamenti e di validarlo in uno scenario il più possibile verosimile. Le esercitazioni permettono di valutare e migliorare l'efficienza di un piano ai diversi livelli: dal numero di risorse in campo al loro coordinamento, dalla simulazione di alcune procedure operative al test di eventuali innovazioni o proposte di modifica.

1.3.4 Informazione alla Popolazione

Requisito fondamentale per un Piano efficace è la conoscenza del piano stesso da parte della popolazione. Il Metodo Augustus propone tre parole chiave per regolare il rapporto tra Piano di emergenza e popolazione:

- **Conoscenza:** corretto uso dei media al fine di rendere edotta la popolazione sulle caratteristiche tecnico-scientifiche del territorio
- **Coscienza:** sforzo da parte delle istituzioni per rendere cosciente la popolazione della presenza di un rischio o della possibilità del verificarsi di eventi che possano mettere a repentaglio il normale svolgersi della vita umana.
- **Autodifesa:** l'informazione e l'addestramento della popolazione coinvolta in uno scenario di rischio sui comportamenti da mettere in atto al verificarsi dell'evento.

1.4 Gestione delle Risorse

Uno degli aspetti fondamentali sottolineati da Galanti nel documento di presentazione del metodo Augustus è l'introduzione del concetto di risorse, ovvero di quegli strumenti, o di quel personale che può essere attivato al verificarsi dell'evento. La risorsa è lo strumento con il quale l'ente di competenza dà una risposta all'avvenimento calamitoso nel tentativo di riportare alla normalità la situazione nel più breve tempo possibile.

1.4.1 La disponibilità delle Risorse

È evidente come una corretta gestione delle risorse sia un aspetto fondamentale nella creazione di un piano di emergenza credibile ed aderente alla realtà. Particolarmente delicato è il concetto di disponibilità delle risorse, ovvero la possibilità di attivarle in un dato momento per un dato compito. Come accennato più volte, molto spesso le risorse non mancano dal punto di vista "fisico", ma mancano le procedure per attivarle o, ancor più grave, manca la conoscenza stessa della risorsa e della sua eventuale disponibilità.

Conoscere quanti aerei per la lotta agli incendi sono disponibili presso un dato aeroporto è condizione senza la quale risulterà complesso stendere un piano efficiente per la lotta agli incendi in una zona boschiva. La disponibilità ha poi la peculiarità di essere particolarmente variabile per alcuni tipi di risorse, in quanto gli stessi aerei già citati potrebbero trovarsi in un dato giorno o settimana presso un altro aeroporto per operazioni di manutenzione o perché impegnati in un'altra missione. La mancanza di tali risorse dovrà, in una corretta gestione di un piano di emergenza, essere ripianata dall'allocazione di altre risorse o dovrà comunque essere fatto noto a coloro i quali, nel caso del verificarsi dell'evento, dovranno prendere decisioni ed assumersi responsabilità con il supporto del piano di emergenza precedentemente realizzato.

1.4.2 La Pre-Allocazione di Risorse

All'atto della realizzazione di un piano di emergenza si predispone una specifica pre allocazione delle risorse basata sulla necessità stimata delle risorse stesse all'atto del verificarsi dell'evento. Tale pre allocazione non impegna le risorse ma risulta essere un dato importante per monitorare l'efficienza stimata del piano stesso d'emergenza. Se la disponibilità di una data risorsa è inferiore al valore della pre allocazione, occorre che tale informazione arrivi all'operatore umano il quale, secondo competenze proprie del dominio metterà in campo le contromisure adeguate.

Capitolo 2

Tecnologie IT a supporto delle Attività di Gestione delle Emergenze

Scopo del capitolo è presentare i principali strumenti utilizzati ad oggi nella gestione delle emergenze. Sono presentati le tecnologie GIS con le loro peculiarità e le reti di sensori. Il capitolo si conclude con un'analisi della Sala di Controllo al fine di evidenziare alcune carenze a livello di interoperabilità e definire alcuni percorsi di ricerca

2.1 GIS: Geographic Information System

Un Sistema Informativo Geografico (Geographic Information System, GIS) è composto dall'hardware, il software e i dati per catturare, gestire, analizzare e mostrare tutti i tipi di informazioni geo referenziate. Secondo la definizione di Burrough (1986) "il GIS è un potente set di strumenti for acquisire, archiviare, estrarre e rappresentare dati spaziali dal mondo reale". Tale definizione nel corso degli anni ha avuto diverse revisioni e si è arrivati ad includere nel GIS anche l'insieme delle persone e delle procedure operative che permettono il funzionamento di tali sistemi: "Il complesso di uomini, strumenti e procedure che permettono l'acquisizione e la distribuzione dei dati nell'ambito

dell'organizzazione e che li rendono disponibili nel momento in cui sono richiesti a chi ne ha la necessità per svolgere una qualsivoglia attività” (Mogorovich P., 1988).

2.1.1 Ambiti d'uso e funzionalità di un GIS

Il GIS permette all'utente di visualizzare, comprendere, ricercare ed interpretare i dati spaziali con diverse modalità, al fine di rivelare relazioni, schemi e trend nella forma delle mappe, dei report o dei diagrammi. Scopo del GIS è quindi, oltre alla rappresentazione, il mettere in relazione tra loro dati diversi, sulla base del loro comune riferimento geografico in modo da rendere possibile la creazione di nuovi dati o insieme di dati, partendo dalla loro rappresentazione.

Esempi di elaborazioni effettuate da un GIS sono:

- **Overlay topologico:** elaborazione basata sulla sovrapposizione di elementi o layer al fine di creare una nuova informazione spaziale
- **Query geo-spaziali:** interrogazioni di basi dati basate su criteri spaziali di localizzazione, inclusione o prossimità
- **Networks analysis:** algoritmi per la creazione di cammini, minimi e non, tra due punti sulla mappa, grazie all'utilizzo di reti di elementi lineari
- **Analisi geo statistiche:** processi ed algoritmi volti ad analizzare l'eventuale correlazione tra dati o variabili geo riferite.
- **Buffering:** creazione di poligoni o forme geometriche sulla base degli attributi di elementi presenti sulla mappa

Il GIS in sostanza è fusione della cartografia, analisi statistica e tecnologia database. Il sistema offre ampie possibilità di interazione con l'utente e un insieme di strumenti che ne facilitano la personalizzazione e l'adattamento alle problematiche specifiche dell'utente mettendo in relazione tra loro dati diversi.

I sistemi GIS sono utilizzati in svariati abiti applicativi come ambiti di business (banche, assicurazioni, logistica, media), di pubblica amministrazione (governo nazionale, locale, protezione civile, difesa militare, gestione emergenze e disastri, polizia, salute,

trasporti), di didattica e scienza (ricerca, librerie e musei, università), ambientali, ambiti relativi alle risorse naturali e servizi di pubblica utilità come elettricità, gas, telecomunicazioni.

2.1.2 Tecnologie e Tecniche GIS

Le moderne tecnologie GIS utilizzano informazioni digitalizzate, create con diversi metodi che vanno dalla digitalizzazione di mappe cartacee, utilizzo di strumenti CAD o acquisizione dei dati via satellite o aerea.

Per una corretta rappresentazione dei dati in un sistema informatico occorre formalizzare un modello rappresentativo flessibile che si adatti ai fenomeni reali. Nel GIS abbiamo tre tipologie principali di informazioni:

- **Geometriche:** informazioni relative alla rappresentazione cartografica degli oggetti rappresentati; quali la forma (punto, linea, poligono), la dimensione e la posizione geografica
- **Topologiche:** riferite alle relazioni reciproche tra gli oggetti (connessione, adiacenza, inclusione ecc...).
- **Informative:** riguardanti i dati (numerici, testuali ecc...) associati ad ogni oggetto.

Il GIS può prevedere l'utilizzo di database relazionali per la gestione delle informazioni.

L'informazione territoriale può essere codificata in un sistema informativo geografico attraverso due tipologie principali di dato: il dato vettoriale (dati costituiti da elementi semplici quali punti, linee e poligoni, codificati e memorizzati sulla base delle loro coordinate) e il dato raster (matrice di celle, generalmente di forma quadrata o rettangolare dette pixel a cui sono associate le informazioni relative a ciò che esso rappresenta sul territorio).

L'aspetto che caratterizza il GIS è l'informazione geometrica attraverso la memorizzazione della posizione del dato impiegando un sistema di proiezione reale che definisce la posizione geografica dell'oggetto. La posizione dell'oggetto può essere definita sulla base dei diversi metodi e modelli di proiezione e di riferimento.

Forte punto di rottura con la cartografia tradizionale su carta è il concetto di scala. La scala di un GIS è infatti un parametro di qualità del dato in quanto è rapportato al numero di cifre significative utilizzate per definire la localizzazione sulla base delle coordinate geografiche.

2.1.3 Gli sviluppi del GIS, gli standard e il web

Molte discipline scientifiche e non hanno o possono beneficiare della tecnologia GIS. Il mercato GIS, particolarmente attivo negli ultimi anni ha avuto come effetto la riduzione dei costi e continui miglioramenti delle componenti hardware e software del sistema. Questi sviluppi hanno portato quindi ad un maggiore uso della tecnologia al di fuori degli ambiti puramente scientifici, di business e industriali.

Stanno inoltre nascendo, a partire dai sistemi GIS, servizi basati sulla posizione (LBS location-based services) i quali permettono attraverso l'utilizzo di device mobili dotati di ricevitore GPS la visualizzazione di punti d'interesse vicini al luogo in cui l'utente effettivamente si trova in un particolare istante..

La continua evoluzione di questi sistema ha necessariamente dovuto tener conto della necessità di uno standard di rappresentazione dei dati spaziali e la loro fruizione via web.

Open Geospatial Consortium

L' Open Geospatial Consortium (OGC) è un consorzio internazionale composto da più di 300 società, agenzie governative e università che partecipano collaborativamente allo sviluppo di specifiche per sistemi di elaborazione di informazioni geospaziali. Ad esempio l'insieme di interfacce aperte e i protocolli definiti nelle specifiche OpenGIS supportano soluzioni interoperabili che permettono la fruizione dei servizi geografici a piattaforme Web, wireless, a servizi location-based e che permettono agli sviluppatori la creazione di servizi spaziali sempre più complessi e fruibili da qualsiasi tipo di applicazione.

I protocolli OGC includono anche il Web Map Service (WMS) e il Web Feature Service (WMF) i quali rappresentano le tecniche di rappresentazioni di informazioni

geografiche attraverso file immagine, la richiesta e l'importazione da parte di client dei dati geografici ed il loro scambio.

Negli ultimi anni si è assistito ad una vera e propria esplosione dell'utilizzo di GIS anche in ambienti non tecnici grazie al facile accesso alle informazioni anche su dispositivi mobili. Grande merito di questa espansione tecnologica è da imputare alle diverse applicazioni di consultazione di mappe su web come ad esempio *Google Maps*. Questi siti web permettono l'accesso ad un' enorme quantità di dati geografici e alcuni produttori software hanno inoltre messo a disposizione delle API che permettono la creazione di applicazioni personalizzate.

2.2 Sensori e Sistemi di Monitoraggio

Il processo di gestione dell'emergenza nasce da un'attenta analisi del territorio e dei rischi ad esso associati. Lo sviluppo tecnologico ha reso disponibile un sempre maggiore utilizzo di sensori per il monitoraggio di dati fisici e di fenomeni naturali. La creazione di reti di sensori ha permesso sistemi di monitoraggio maggiormente efficaci e distribuiti sul territorio, introducendo alcune difficoltà legate alla comunicazione, alla topologia ed allo scambio di informazioni tra sensori differenti per tecnologia, linguaggi ed ubicazione.

La definizione di Wireless Sensor Network ha risolto le maggiori difficoltà legate alla gestione di oggetti dotati di energia, memoria e capacità di calcolo limitate, strutture di reti con topologie variabili e sistemi di comunicazione e di indirizzamento dedicati.

I sistemi informativi di gestione dell'emergenza fanno un uso intensivo di dati sensoriali.

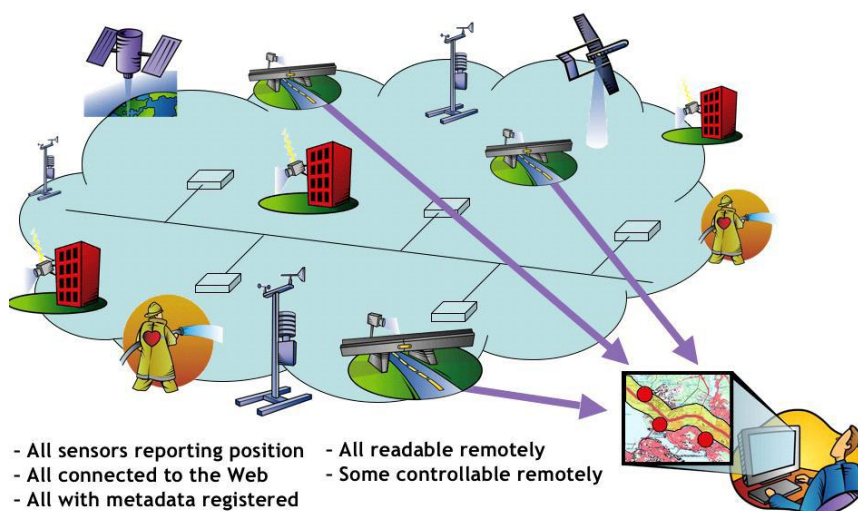


Figura 2 - Sensor Web (OGC White Paper – OGC Sensor Web Enablement: Overview And High Level Architecture.)

Emersa già da qualche anno la necessità di definire architetture di sensori con un alto livello di astrazione ed indipendenti dalle singole soluzioni tecnologiche a livello di comunicazione e rilevamento dei dati, la ricerca ha prodotto alcuni standard per la definizione di reti di sensori interconnessi attraverso tecnologie web, accessibili o controllabili da remoto.

2.3 Sale di Controllo

Le sale di controllo e le sale operative svolgono un ruolo centrale nel processo di gestione dell'emergenza e delle attività ad essa collegate. Sul territorio nazionale sono diverse le strutture di diverso livello operative anche in H24. Le sale di controllo possono essere considerate il centro decisionale e pensante dell'organizzazione di Protezione Civile e svolgono sia attività legate alle emergenze al verificarsi dell'evento sia attività di monitoraggio e prevenzione e, all'interno delle più avanzate, si possono riconoscere i tratti distintivi del metodo Augustus, quale la presenza fisica delle funzioni di supporto.

2.3.1 Sistema Informativi ed integrazione dei Dati

Le informazioni descritte in questo paragrafo derivano da un'analisi svolta anche sul campo grazie alla visita alla Sala Regionale Lombarda.

A livello di sistema informativo le Sale di Controllo dispongono di uno o più strumenti basati su GIS utilizzati per le funzioni più diverse, ma soprattutto per il monitoraggio in tempo reale del territorio (controllo scariche e fulmini in tempo reale, situazione meteo ed idrica etc...). Nonostante una buona dotazione di informazioni e di strumenti con cui trattarle, si evidenzia una certa carenza a livello di integrazione dei dati, processi ed attori coinvolti all'interno delle Sale.

Integrazione di Dati e Sensori

Come presentato in precedenza le reti di sensori si stanno rapidamente affermando come uno dei sistemi cardine dei processi di gestione del territorio. Le Sale hanno accesso a diversi network di sensori e ad una grande moltitudine di dati i quali però risultano spesso poco interoperabili a causa dell'utilizzo di strumenti proprietari a livello hardware, ma soprattutto software, e ad una reale mancanza di standard di integrazione.

Integrazione tra le diverse funzioni all'interno della Sala

Nel metodo Augustus le funzioni di supporto ricoprono un ruolo fondamentale per una corretta gestione dell'emergenza ognuna secondo le proprie competenza. All'interno delle sale Provinciali/Regionali la presenza di incaricati delle diverse funzioni rappresenta una forte opportunità di coordinamento e di organizzazione che viene però solo parzialmente supportata da un'integrazione tra i sistemi informativi delle diverse funzioni e tra le funzioni e i GIS della Sala di Controllo

Gestione del flusso delle informazioni in entrata ed in uscita

La gestione dei flussi di informazioni in entrata ed in uscita dalle Sala di Controllo o all'interno della Sala stessa richiede spesso l'intervento umano come "middleware" per la loro interpretazione (nel caso di dati appartenenti ad uno specifico dominio di rischio) o per la loro distribuzione tramite telefono, radio o e-mail.

Integrazione tra Sale di Controllo

Estendendo il concetto di interoperabilità, occorre valutare la possibilità di scambiare dati ed informazioni tra Sale di Controllo sia sullo stesso livello organizzativo sia su livelli differenti soprattutto nel caso di rischi od emergenze ampie e che coinvolgono strutture diverse per competenze e localizzazione. In questo caso l'integrazione è spesso imposta dall'alto attraverso l'adozione di sistemi omogenei tra le diverse Sale che possono andare a sovrapporsi o sostituire sistemi o tecnologie già provate e sperimentate nelle singole Sale.

Capitolo 3

Interoperabilità negli Emergency Management Information System

Scopo del capitolo è descrivere lo stato dell'arte attraverso i principali standard ad oggi esistenti o in via di definizione che contribuiscono a creare i presupposti per la definizione di sistemi interoperabili.

Si presentano standard differenti, dagli standard più generici (Web Service e Architetture SOA) a i progetti più legati al dominio degli EMIS. Si presentano inoltre alcune soluzioni già esistenti per l'interoperabilità di EMIS

3.1 Gli standard operativi

Il panorama dei sistemi informativi per la gestione del rischio è costellato di innumerevoli applicazioni stand-alone che assolvono solamente un determinato compito senza preoccuparsi di mettere a conoscenza dei risultati ottenuti altri utenti o processi limitando le potenzialità computazionali all'interno della singola macchina o sistema informativo.

La possibilità di scambiare informazioni, primo passo verso l'integrazione e l'interoperabilità, tra diversi sistemi passa attraverso la difficile definizione delle modalità di comunicazione tra i sistemi stessi: la definizione e l'utilizzo di standard operativi condivisi risulta da sempre la giusta soluzione al problema.

Uno Standard è un insieme di norme, raccomandazioni o specifiche prestabilite da un'autorità e riconosciute tali con lo scopo di rappresentare una base di riferimento o paradigma codificato per la realizzazione di tecnologie fra loro compatibili e interoperabili, che siano componenti hardware, software o infrastrutture di rete. Nell'ambito prettamente informatico diversi enti internazionali si fanno carico della specifica degli standard come l'ISO (International Organization for Standardization), l'IEEE (Institute of Electrical and Electronics Engineers), il W3C (World Wide Web Consortium), il consorzio OASIS o anche l'OGC (Open Geospatial Consortium) che si occupa di definire le specifiche tecniche per i servizi di geospaziali e di localizzazione.

Grazie al forte sviluppo delle tecnologie GIS e alla ricerca legata alla gestione condivisa del rischio molti standard sono stati definiti nel tentativo di soddisfare l'esigenza di interoperabilità. Se nel campo della gestione di dati geolocalizzati la ricerca ha definito da qualche anno standard condivisi e riconosciuti universali, in materia di gestione del rischio non esiste ancora una standardizzazione diffusa, bensì alcuni progetti, proposte o standard giovanissimi che ancora debbono affermarsi. Nei seguenti paragrafi saranno brevemente presentati i più interessanti ai fini della ricerca. (Oasis Standard)

3.2 Servizi Web e Catalogazione

All'interno del panorama complesso ed articolato delle SOA, si desidera porre l'attenzione su due aspetti ritenuti fondamentali nel progetto di integrazione di EMIS: i servizi web e la loro catalogazione.

3.2.1 Web Service

I web service, unità fondamentale delle architetture orientate ai servizi (SOA) sono un tipo di applicazioni che permettono lo scambio di dati attraverso messaggi indipendenti dalla particolare implementazione applicativa. Presenti sul mercato da diversi anni hanno contribuito a facilitare la comunicazione e lo scambio di informazioni tra sistemi informativi diversi per architettura ed implementazione.

Nelle implementazioni odierne lo scambio di informazioni attraverso servizi web si attua grazie alla pubblicazione da parte del servizio di un “contratto di servizio” (WSDL) che definisce le operazioni e le funzionalità che il servizio web offre e lo scambio di messaggi XML secondo lo standard SOAP via protocollo HTTP tra client e servizio web. (Papazoglou, 2007)

3.2.2 Servizi Catalogo OGC Catalog e UDDI

In un’architettura orientata ai servizi un ruolo fondamentale è riservato alle funzionalità di ricerca e pubblicazione dei diversi servizi e delle risorse accessibili attraverso l’architettura.

In quest’ottica occorre definire un oggetto che funga da Catalogo dei Servizi attraverso la creazione di registri attraverso i quali ricercare e scoprire i servizi disponibili, le funzionalità messe a disposizione e il “linguaggio parlato” da ciascun servizio. Da questi semplici concetti, negli ultimi anni sono state diverse le soluzioni trovate per la definizione di cataloghi di servizi e di standard per la definizione di meta informazioni.

Sono stati individuati i due seguenti standard:

OGC Catalogue Service Specification: specifica largamente accettata nell’ambito dei dati geospaziali, descrive le interfacce per ricercare e pubblicare informazioni descrittive su dati e servizi. In aggiunta alla specifica più generale, sono disponibili sul mercato diversi profili applicativi che raffinano lo standard con implementazioni specifiche per specifici metadati (ad esempio OGC ISO19115/ISO19119 Application Profile for CSW 2.0)

UDDI: l'obiettivo del "the Universal Description, Discovery, and Integration" (UDDI) è la definizione di metodi standard per la pubblicazione e la scoperta di servizi esistenti all'interno di una SOA. UDDI, sviluppata dal consorzio OASIS, sebbene non abbia avuto un grosso successo come registry globale è comunque utilizzato in ambienti controllati come registry di servizi sviluppati con tecnologia Web Service.

3.3 Geolocalizzazione e Sensori

Come descritto nel capitolo 2, gran parte del supporto IT nei processi di gestione dell'emergenza è fornito da strumenti per la manipolazione di dati geo referenziati come dati cartografici o dati provenienti da reti di sensori. Si descrivono di seguito alcuni dei principali standard sviluppati per supportare la condivisione e lo scambio di tali informazioni.

3.3.1 WMS - Web Map Service

Lo standard WMS, sviluppato dall'OGC e divenuto standard ISO nel 2005, permette la definizione di servizi dedicati alla produzione di mappe georeferenziate. Un Web Map Service è un fornitore di dati georeferenzati utilizzabili da client HTTP attraverso l'accesso ad un determinato URI. I dati restituiti da un WMS sono essenzialmente dati grafici sotto forma di dati raster (JPEG, PNG, GIF) e dati vettoriali come nel caso di SVG permettendo così la sovrapposizione o la visualizzazione di mappe o elementi cartografici sovrapposti.

Lo standard mette a disposizione del client 3 funzionalità principali

- **GetCapabilities:** ritorna i metadati associati al servizio descrivendo i dati e le mappe disponibili.
- **GetMap:** restituisce la mappa sulla base dei parametri richiesti (sovrapposizione di layer, formato dati, gestione trasparenze...)

- **GetFeatureInfo:** metodo opzionale che permette di ottenere informazioni sugli oggetti presenti all'interno della mappa

WMS gode di un forte supporto grazie anche all'implementazione della specifica da parte di software e GIS proprietari oltre allo sviluppo di diversi tool da parte della comunità OpenSource. (OGC, 2006)

3.3.2 WFS – Web Feature Service

La specifica WFS permette la definizione di servizi simili alla specifica WMS ponendo però maggiore attenzione al dato che nel caso di Web Feature Service non risulta essere una o più immagini raster, bensì un dato strutturato ed indipendente dalla particolare cartografia. WFS permette quindi la distribuzione di veri e propri oggetti geo referenziati descritti attraverso lo standard GML (Geography Markup Language), una grammatica basata su XML definita dall'OGC.

La specifica WFS è in questi mesi la protagonista di un processo di revisione in previsione dell'uscita della specifica 2.0 che dovrebbe rendere lo standard ancor più completo e maturo. La specifica 1.1 mette a disposizione dell'utilizzatore funzionalità di Query basata su tipologia o dati spaziali, di Lock delle Feature per la modifica, creazione e cancellazione delle stesse, di descrizione del servizio e delle singole Feature.

Le funzionalità di maggiore interesse in ottica EMIS sono:

- **GetCapabilities:** restituisce un descrittore del servizio e dei tipi di feature disponibili tramite documento XML
- **GetFeature:** restituisce le feature disponibili e risultanti dai parametri di query forniti al servizio. La query può essere eseguita su diversi parametri, dai semplici identificativi, alla tipologia di feature, alla loro posizione o alla loro appartenenza ad una determinata area (bounding-box)
- **DescribeFeatureType:** restituisce la descrizione dettagliata della struttura delle feature offerte dal servizio.

Si riporta di seguito alcuni frammenti di Codice relativi ad una ricerca di feature di tipo Roads o Rails presenti in una determinata area. La gestione di Feature (siano essi punti vulnerabili, scuole o risorse utilizzabili in caso di emergenza) è infatti uno degli aspetti di maggiore interesse nella gestione di Scenari di rischio.

Estratto della Richiesta di Feature di tipo Roads o Rails all'interno dell'area di interesse

```
<GetFeature version="1.1.0" service="WFS" handle="Example Query" [...]>
  <Query typeName="myns:ROADS">
    <wfs:PropertyName>myns:path</wfs:PropertyName>
    <wfs:PropertyName>myns:lanes</wfs:PropertyName>
    <wfs:PropertyName>myns:surfaceType</wfs:PropertyName>
    <ogc:Filter>
      <ogc:Within>
        <ogc:PropertyName>myns:path</ogc:PropertyName>
        <gml:Envelope srsName="EPSG:63266405">
          <gml:lowerCorner>50 40</gml:lowerCorner>
          <gml:upperCorner>100 60</gml:upperCorner>
        </gml:Envelope>
      </ogc:Within>
    </ogc:Filter>
  </Query>
  <Query typeName="myns:Rails">
    <wfs:PropertyName>myns:track</wfs:PropertyName>
    <wfs:PropertyName>myns:gauge</wfs:PropertyName>
    <ogc:Filter>
      <ogc:Within>
        <ogc:PropertyName>myns:track</ogc:PropertyName>
        <gml:Envelope [...] />
      </ogc:Within>
    </ogc:Filter>
  </Query>
</GetFeature>
```

Estratto della Risposta di Feature di tipo Roads o Rails all'interno dell'area di interesse

```
<wfs:FeatureCollection [...] >
```

```

<gml:featureMember>
  <Roads gml:id="Roads.100">
    <path>
      <gml:LineString gid="1" [...]>
        <gml:posList>10 10 10 11 10 12 10 13</gml:posList>
      </gml:LineString>
    </path>
    <surfaceType>ASPHALT</surfaceType>
    <nLanes>4</nLanes>
  </Roads>
</gml:featureMember>
<gml:featureMember>
  <Roads gml:id="Roads.105">
    <path>
      <gml:LineString gid="2" [...]>
        <gml:posList>10 10 10 11 10 12</gml:posList>
      </gml:LineString>
    </path>
    <surfaceType>GRAVEL</surfaceType>
    <nLanes>2</nLanes>
  </Roads>
</gml:featureMember>
<gml:featureMember>
  <Rails gml:id="Rails.119">
    <track>
      <gml:LineString gid="n" [...]>
        <gml:posList>15 10 16 11 17 12</gml:posList>
      </gml:LineString>
    </track>
    <gauge>24</gauge>
  </Rails>
</gml:featureMember>
</wfs:FeatureCollection>

```

3.3.3 SWE – Sensor Web Enablement

Sensor Web Enablement è una raccolta di standard sviluppati dall'OGC al fine di rendere disponibili strumenti e metodologie per la ricerca e lo scambio di informazioni

relative a sensori e a reti di sensori (Mike Botts, 2007). L'architettura SWE nasce dall'astrazione di sensore come servizio inserito in un'architettura SOA in grado di monitorare eventi e fenomeni di diverso tipo. L'OGC identifica gli standard sviluppati all'interno di SWE con la stessa finalità abilitante che ha avuto l'HTML e la definizione del protocollo HTTP per lo scambio di informazioni in ambiente WEB.

Perseguendo l'obiettivo di poter implementare reti di sensori eterogenei, scalabili e fortemente interoperabili tra loro l'OGC ha definito alcune funzionalità ritenute la base delle future implementazioni:

- Scoperta di sistemi di sensori, delle loro osservazioni e dei processi di osservazione che combacino con le richieste dell'utente.
- Determinazione delle capacità di un sensore e della qualità delle sue misurazioni
- Accesso ai parametri e alle informazioni di un sensore al fine di determinare in modo automatico la sua geo-localizzazione
- Recupero di dati in tempo reale o di serie di dati rilevati in un intervallo di tempo attraverso la definizione di codifiche standard
- Creazione di attività o task al fine di ottenere le misurazioni ritenute rilevanti
- Sottoscrizione e conseguente pubblicazione di messaggi di allerta o allarmi creati da sensori o reti di sensori sulla base di criteri definiti

Al fine di raggiungere lo scopo di permettere l'accesso, il recupero dei dati e la gestione di sensori eterogenei attraverso la rete internet, creando una sorta di architettura "Plug-and-Play", all'interno di SWE sono stati definiti alcuni standard per la codifica e lo scambio di informazioni basate sul meta-linguaggio XML:

Observation & Measurements Schema (O&M)

Specifica per la creazione di modelli standard e loro traduzione in linguaggio XML per la codifica di osservazioni e misurazioni di un sensore sia in real-time che precedentemente rilevate

Sensor Model Language (SensorML):

Specifica di modelli e descrizione XML standard a supporto della descrizione di sistemi di sensori e processi. SensorML fornisce inoltre le informazioni per la scoperta, la descrizione e localizzazione di sensori e misurazioni. SensorML fornisce infine la possibilità di rendere in dettaglio i componenti low-level dei sensori in modo tale da poter descrivere anche il processo con cui le misurazioni delle osservazioni vengono effettuate.

Transducer Markup Language (TML)

Metodo e codifica a supporto della cattura, dello scambio e dell'archiviazione di dati da e verso i sistemi di sensori. TML definisce un set di modelli per la descrizione le caratteristiche hardware della risposta. La necessità deriva dalla constatazione che molto spesso il valore rilevato è una funzione di comportamenti o effetti fisici di altra natura (es. fenomeno elettrico per la misurazione di una temperatura).

Sensor Observation Service (SOS)

Definisce una metodologia per la pubblicazione e l'accesso alle informazioni e alle misurazioni effettuate dalla rete di sensori. E' forse lo standard più importante definito all'interno del SWE in quanto permette l'accesso alle osservazioni di sensori eterogenei e differenti tra loro definendo una metodologia che permetta l'astrazione da sensore a "servizio fornitore di informazioni"

Sensor Planning Service (SPS)

Tipologia di servizio che permette ai client di pianificare la richiesta di una particolare collezione di dati per uno o più sensori

Sensor Alert Service (SAS)

Servizio per la gestione di messaggi di Alert. L'idea di SAS è la creazione di una sorta di registro (e non un servizio di messaggistica) in cui pubblicare le informazioni circa i protocolli e la natura degli allarmi offerti. SAS intende conservare una forte interoperabilità con standard e strumenti già esistenti. Ad esempio, attraverso SAS un

utente può registrarsi per ricevere CAP Alert (il protocollo CAP sarà descritto in seguito)

Web Notification Services (WNS)

Definizione di servizio attraverso cui un client può gestire lo scambio di messaggi asincroni con uno più servizi

3.3.4 CAP – Common Alerting Protocol

Il Common Alerting Protocol (CAP) è un formato di dati basato su XML per lo scambio di avvisi di emergenza e allerte pubbliche su tutti i tipi di canali. La specifica CAP 1.2 è stata definita dal consorzio OASIS all'inizio dell'anno 2010. CAP non predilige alcuna particolare applicazione o tipo di telecomunicazione ed è compatibile con le tecniche emergenti come Web Services o Commercial Mobile Alert System (CMAS) ma anche retro compatibile con i formati di alert esistenti come il SAME (Specific Area Message Encoding) usato negli Stati Uniti in Weatheradio per la trasmissione da parte degli Emergency Alert System.

Il CAP permette la distribuzione di un messaggio di avviso simultaneamente attraverso diversi sistemi di allerta aumentando l'efficacia dell'avviso stesso. CAP semplifica inoltre il compito di attivazione dei responsabili dell'emergenza.

Uno dei benefici chiave del CAP è la riduzione dei costi e della complessità operativa con l'eliminazione del bisogno di svariate interfacce software per le diverse fonti e canali di allerta. Il formato di messaggio CAP è inoltre utilizzato e supportato anche da diversi tipi di sensori, in un'ottica di WSN basata su architettura SOA. (Oasis Standard)

3.3.5 Un esempio di Implementazione: FrameWork MapServer

Si presenta ora brevemente il framework Open Source MapServer (MapServer), uno strumento ampiamente utilizzato nella gestione di dati geolocalizzati.

Come già anticipato, nei sistemi di gestione del rischio assumono un ruolo centrale i dati che riguardano la posizione nello spazio sia di risorse che luoghi coinvolti in uno scenario di rischio; è possibile riferirsi a tali entità in modo preciso e sistematico attraverso i dati geospaziali salvati in basi dati dedicate.

Se i dati geospaziali rappresentano il formato di elaborazione dell'informazione spaziale, la cartografia rappresenta il più naturale accesso a tali dati da parte dell'utente umano.

MapServer permette la rappresentazione di questi dati geospaziali: MapServer è un motore grafico di rendering dei dati geospaziali che dall'estrapolazione di dati geolocalizzati crea file "mappa".

La forza di MapServer sta nella sua flessibilità e per l'aderenza agli standard: infatti il motore permette di pubblicare servizi web conformi all'Open Geospatial Consortium utilizzando specifiche già descritte come WMS, WFS e WCS ovvero Web Map Service, Web Feature Service, Web Coverage Service che rappresentano tecniche di rappresentazioni di informazioni geografiche attraverso file immagine, la richiesta e l'importazione da parte di client dei dati geografici ed il loro scambio.

In un'ottica di integrazione dei servizi MapServer si configura come la possibile soluzione all'accesso e alla visualizzazione di dati spaziali conformi agli standard in modo condiviso.

3.4 Interoperabilità di EMIS

Si descrivono di seguito alcuni progetti o standard che hanno gettato alcune basi importanti nella definizione di architetture per l'interoperabilità di EMIS.

3.4.1 Progetto Orchestra

ORCHESTRA (an Open Architecture and Spatial Data Infrastructure for Risk Management) è il più grande progetto integrato del Sixth Framework Programme (FP6) e si concentra sulle sfide tecnologiche che limitano una gestione efficace delle informazioni in materia di rischio ambientale.

Il progetto è iniziato nel Settembre del 2004 e terminato in Febbraio 2008 e vi ha partecipato un consorzio di 14 partner in Europa e ha ottenuto un finanziamento di 8,2 Milioni di € a fronte di 13,7 M€ stimati come budget.

Il progetto ORCHESTRA inizia i suoi lavori a partire dall'analisi delle attività di gestione di rischio che, come confermato dagli ultimi eventi calamitosi, devono sempre più far fronte a diversi eventi, sia naturali che causati dall'uomo, ed alla loro concatenazione ponendo l'attenzione a scenari multi rischio.

Le attività di Gestione del Rischio, a livello Europeo, coinvolgono una serie di organizzazioni operanti a diversi livelli amministrativi e dotate di propri sistemi e servizi. Come conseguenza di questa complessità, la condivisione di informazioni richiesta per la Gestione del Rischio ambientale è spesso limitata ad uno scambio di dati non elaborati e non strutturati, determinando una serie di problemi di natura sintattica e semantica (es. il semplice utilizzo di idiomi diversi: italiano, francese, inglese).

Nella maggior parte dei casi, l'efficienza è limitata da fattori che da una parte sono legati a confini amministrativi e legali e dall'altra alla mancanza di interoperabilità dal punto di vista tecnico.

L'applicazione di molteplici politiche, procedure, standard e sistemi di natura diversa tra loro, comporta problemi di coordinamento nell'analisi dei dati, nella fruizione dell'informazione e nella gestione delle risorse, tutti fattori critici nella Gestione del Rischio.

Alla luce di questo ORCHESTRA si pone come obiettivo principale la progettazione e modellazione di una architettura orientata ai servizi basata su standard de-facto e de-iure in grado di migliorare l'interoperabilità tra gli attori coinvolti in attività di Gestione del rischio ambientale e la validazione dell'architettura proposta in scenari multi rischio, sviluppando una serie di servizi utili per gli utenti finali.

Il principale risultato del progetto ORCHESTRA è quindi la definizione di un Modello di Riferimento per una Architettura "ORCHESTRA" denotato come RM-OA. RM-OA è conforme allo standard ISO/IEC 10746 RM-ODP, in particolare ai cosiddetti 5 "viewpoints" del "Reference Model for Open Distributed Processing" (Modello di Riferimento per l'Elaborazione in Ambiente Aperto Distribuito), nel contesto di una architettura orientata ai servizi.

Il modello RM-OA segue fondamentalmente un approccio suddiviso in due passi. Il primo passo ha come focus la specifica combinata (generica e indipendente dalla piattaforma di sviluppo) dei due viewpoints Information e Service ed è basata sui requisiti definiti nel viewpoint Enterprise. Il termine "generico" indica che l'Architettura ORCHESTRA è indipendente dal dominio applicativo e da una specifica organizzazione ambientale avente una data localizzazione. In tal modo, essa può essere utilizzata per molti altri modelli di business, non necessariamente legati ai domini ambientale. Per rendere l'architettura indipendente dalla piattaforma di sviluppo essa è stata specificata utilizzando UML.

Queste specifiche definite in UML sono definite "astratte" poiché indipendenti da una particolare infrastruttura di servizi (per esempio, W3C Web Services), aspetto di particolare rilevanza se si tiene conto che queste infrastrutture di servizi di norma cambiano nel giro di una decade (OSFDCE, CORBA, Web Services). Nel prossimo capitolo, sono illustrati gli aspetti implementativi relativi ai "viewpoints" Technology and Engineering delle specifiche generiche adottate.

I principali servizi descritti nell'architettura sono:

- Ricerca, navigazione e accesso a informazioni attraverso sistemi diversi già operanti
- Accesso a servizi offerti da altre organizzazioni
- Interoperabilità semantica
- Facilità di integrazione di informazioni fornite in applicazioni finali, come sistemi di supporto alle decisioni

Il progetto Orchestra si è concluso con la definizione di alcune best practice accettate anche dall'OGC. Il progetto contiene però alcuni punti da approfondire ulteriormente quali la possibilità di gestire in modo automatico l'aggiornamento delle informazioni senza ricorrere a elaborazioni Batch o la mancanza di un oggetto riconducibile allo "Scenario di Rischio". Tali funzionalità non sono precluse dall'architettura particolarmente flessibile e scalabile, ma risultano essere due direzioni di sviluppo interessanti. La definizione di interfacce astratte se da un lato risulta essere un punto di forza, rende l'architettura non immediatamente implementabile a meno delle dovute ipotesi su tecnologie e linguaggi da utilizzare nello scambio di informazioni.

3.4.2 OASIS Fp6

Il progetto OASIS fa parte come Orchestra dell' FP6 e il suo obiettivo è definire e sviluppare un framework IT basato su una architettura aperta e flessibile usando standard sia esistenti che proposti che possa diventare la base dei sistemi di rischio dei disastri e emergenze.

Il progetto è iniziato nel 2004 ed è durato 4 anni.

OASIS nasce con lo scopo di facilitare la cooperazione attraverso l'utilizzo di sistemi informativi a supporto degli enti coinvolti nelle operazioni di soccorso, come la protezione civile, le forze di polizia, i vigili del fuoco etc...

Il progetto fornisce insieme al framework un insieme iniziale di applicazioni che copre i bisogni principali necessari agli utenti finali durante la gestione dell'emergenza. Uno dei fattori principali che OASIS considera è **la necessità di una chiara conoscenza della**

situazione in essere al fine di procedere con operazioni di gestione dell'emergenza efficaci.

La situazione reale all'atto del verificarsi di un evento è definita e astratta come un insieme di informazioni riportate da differenti team coinvolti nella gestione dell'evento, ognuno secondo le proprie competenze; la costruzione di tale quadro si basa sullo scambio di informazioni da parte dei diversi attori coinvolti nelle operazioni.

Per facilitare questi scambi di informazioni il progetto ha creato e definito uno strumento standard che permette di scambiare informazioni dettagliate sulla situazione e ne permette l'interscambio tra i sistemi coinvolti nella gestione dell'emergenza.

Tale strumento è il Tactical Situation Object (TSO) il quale descrive l'evento, le risorse coinvolte e le operazioni in corso in modo preciso e non ambiguo.

Il TSO è stato proposto come standard poiché può descrivere tutti i tipi di evento, le risorse coinvolte nelle operazioni e le operazioni in corso, oltre ad altri dati utili, come il numero di persone coinvolte, il tutto in un unico oggetto dotato di un identificativo e con la possibilità di essere geo-referenziato.

I dati contenuti nel TSO sono codificati in modo tale che siano leggibili sia da un elaboratore che da un essere umano e possano essere scambiati attraverso canali di comunicazione diversi in modo trasparente grazie alla trasmissione in formati, linguaggi e piattaforme differenti.

I principali risultati del progetto OASIS sono stati la definizione di un protocollo e di un linguaggio per lo scambio di informazioni, il TSO, duttile e di facile interpretazione. Il focus del progetto, principalmente votato alla fase di risposta all'emergenza, rende però il TSO un oggetto fortemente legato alla singola fase e non così facilmente esportabile nelle altre fasi di gestione dell'emergenza



Figura 3 - Esempio di TSO

Capitolo 4

Augustus 2.0: Informazione e Scenario in ambiente SOA

Scopo del presente capitolo è l'introduzione di alcuni concetti quali l'applicazione di metodologie tipiche del web 2.0 al supporto IT per Augustus nell'ottica di un ipotetico Augustus 2.0. il capitolo prosegue con la definizione dei requisiti architetturali: informazione come servizio, l'aggiornamento delle informazioni tramite eventi, il concetto di scenario come servizio di servizi e le sue possibili estensioni.

4.1 Web 2.0 Collaborare e Condividere

Il termine Web 2.0 è entrato ormai da tempo nel vocabolario comune per indicare una nuova fase nel rapporto tra WWW e utenti dei servizi che su esso o attraverso trovano l'erogazione. Il termine 2.0 da sempre legato allo sviluppo di software ed in particolare alla gestione delle versioni in questo caso si applica più alla filosofia di fondo che alla nascita di una nuova tipologia di applicativi o strumenti dal punto di vista tecnologico.

Il Web, nella sua fase 2.0, si apre alla collaborazione ed alla condivisione dei contenuti e delle informazioni su esso residenti, rompendo lo schema classico di fornitore ed utilizzatore delle informazioni tipico della carta stampata o della televisione. Il Web diventa una immensa piattaforma per la partecipazione, la collaborazione e la condivisione di informazioni create e gestite da attori diversi, dalle grandi istituzioni a semplici utenti.

Il processo si evidenzia in ambienti come i Blog o Wiki o gli RSS in cui l'informazione viene creata da diversi soggetti, ognuno per le proprie competenze ed inserita all'interno di una piattaforma universale che la rende fruibile a tutti. La tipologia di struttura formata dai collegamenti tra le informazioni si sposta da una topologia a stella ad interconnessioni multiple tra informazioni simili o connesse tra loro attraverso link, tag o social network.

4.1.1 SOA – La collaborazione attraverso i servizi

SOA è una architettura ormai affermata da qualche anno nel panorama dell'interoperabilità tra sistemi informativi e ha visto nel Web 2.0 uno dei principali promotori del proprio sviluppo. In un ambiente fortemente indirizzato verso la collaborazione e la gestione di informazioni condivise ed interconnesse occorre avere un supporto tecnico alla gestione dell'informazione svincolata da paradigmi o linguaggi implementativi. L'interoperabilità garantita dalle soluzioni a servizi, già presentati nel capitolo 3, risulta efficace e di semplice utilizzo in quanto i servizi:

- **sono interoperabili** tra loro e grazie agli standard possono comunicare tra loro indipendentemente dalla tecnologia con cui sono sviluppati.
- **possono essere composti** ed associati per creare nuovi servizi
- **possono essere riutilizzati** in diversi scenari a seconda del loro contratto di servizio

Una volta definita un'architettura di servizi si aprono nuove possibilità legate all'integrazione ed alla distribuzione di informazioni ed elaborazioni non dovendo più

concentrarsi sulla singola comunicazione tra diversi sistemi informativi a livello tecnico-implementativo.

In un ambiente di tale complessità dal punto di vista organizzativo come quello delle sale di Controllo definite nel Metodo Augustus, l'astrazione delle funzionalità e delle informazioni a livello di servizi si rivela un potente strumento. Come esempio immediato, il concetto di sensore stesso risulta ridefinito in una concezione che lo vede come "servizio fornitore di un dato".

4.1.2 Il web 2.0 in scenari d'emergenza

Ridefinendo il concetto di web 2.0 (collaborazione, partecipazione e condivisione) in una modalità più astratta e generale sono nate differenti scuole di pensiero che intendono applicare le peculiarità del 2.0 alla gestione del business, della medicina e dell'apprendimento. Con le dovute premesse occorre riconoscere che tale filosofia può essere applicata e giovare anche alla gestione del rischio e delle emergenze sia in fase di preparazione all'emergenza che durante la gestione dell'emergenza stessa, come alcune esperienze spontanee e recenti ci hanno confermato.

Terremoto in Abruzzo

Il terremoto in Abruzzo è risultato essere il primo evento italiano su grande scala che ha avuto un riflesso molto importante sul web e sulle piattaforme di comunicazione 2.0. A poche ore/pochi giorni dall'evento sono nati sul web diverse iniziative tipiche del social media per una maggiore circolazione delle informazioni. Dai Blog ai Wiki fino ad arrivare all'utilizzo di Twitter come strumento di comunicazione nonostante la bassa penetrazione del portale in Italia. A conferma dell'importanza dello strumento web anche in situazioni estreme occorre riportare la creazione da parte di Google Inc. di un concentratore d'informazione, tuttora attivo, sull'evento sismico e l'iniziativa del Corriere della Sera.it con l'apertura di un forum dedicato che fin da subito si è caratterizzato come luogo in cui scambiarsi informazioni, offerte e richieste di aiuto. (Forum Terremoto Centro Italia)

OGGETTO	INVIATO DA	INVIATO IL
R offro il mio aiuto	fabrizio	06-04-2009 21:22
R offro aiuto	stefano	06-04-2009 21:14
R aiuto	claudio	06-04-2009 20:47
R offerta di aiuto	luciana biondi	06-04-2009 19:47
R aiuto x i terremotati	maria paola	06-04-2009 19:15
R offro aiuto	Bernardi Matteo	06-04-2009 19:05
R Aiuto	Matteo	06-04-2009 16:53
R Richiesta di aiuto	vito giuseppe	06-04-2009 16:53
R vorrei tanto dare aiuto a questa gente!!	Eleonora	06-04-2009 16:15
R DATECI UN CONTO CORRENTE PER DARE IL NOSTRO AIUTO	VALENTINA dalla germania	06-04-2009 16:15
R Aiuto	Federico	06-04-2009 15:35
R Un aiuto subito	Alberto	06-04-2009 15:08
R Offro Aiuto	Giovanni da Milano	06-04-2009 13:45
R aiuto per amici	helga	06-04-2009 13:32
R disponibile ad un aiuto	Ernesto	06-04-2009 13:32
R aiuto	antonella	06-04-2009 12:47
R Offerta aiuto	Marco da Pordenone	06-04-2009 12:47

Figura 4 - Forum Corriere della Sera - Terremoto in Centro Italia

Terremoto Haiti

Nel caso dell'evento sismico registrato ad Haiti nel 2010 l'utilizzo di tecnologie 2.0 è risultato ancor più evidente ed importante. La rete è venuta a conoscenza dell'evento dalla tempestiva segnalazione via Twitter di un abitante di Port-au-Prince, prima ancora delle segnalazioni sulle testate di informazione. Twitter e Facebook sono ben presto risultati essere uno strumento tanto utile quanto destrutturato per offrire e richiedere aiuto o per informare con foto e testimonianze di prima mano. (Frediani, 2010)

4.2 Verso Augustus 2.0

Nel Capitolo 1 si è messo in luce come sia necessario un continuo aggiornamento delle informazioni facenti parte un piano di emergenza o uno scenario di Rischio. L'aggiornamento di tali informazioni, fondamentale per la continua attualità del piano, avviene ad oggi, nella maggior parte dei casi, attraverso procedure di tipo batch a carico del responsabile della singola funzione. La procedura si svolge ad intervalli più o

meno regolari a seconda della predisposizione al cambiamento dell'informazione o dell'oggetto analizzato. Risulta infatti chiaro che alcune informazioni (es. il livello idrometrico di un fiume) sono fortemente dipendenti dal tempo e che l'informazione stessa può risultare molto differente anche a breve distanza, come al contrario alcune informazioni sono poco variabili nel tempo (la presenza di uno snodo ferroviario o la presenza di un'azienda chimica).

4.2.1 Collaborazione

La creazione e la manutenzione di uno Scenario di Rischio, così come la creazione di un Piano di Emergenza, sono processi che coinvolgono diversi attori ognuno secondo le proprie competenze e responsabilità. Il processo è fortemente condiviso e collaborativo, così come lo è la fase di manutenzione e di aggiornamento del piano stesso. Tali presupposti permettono la definizione di una nuova tipologia di sistema informativo che permetta di offrire un supporto tecnologico ai processi di comunicazione e collaborazione già esistenti offrendo benefici in termini di tempo, di tempestività e di controllo del flusso di informazioni. Scopo della nuova tipologia di SI è l'utilizzo di tecniche e tecnologie 2.0 applicate al metodo Augustus il quale, già nel documento di Galanti del 1997, auspicava forme di collaborazione ai diversi livelli operativi introducendo il concetto di Piano di emergenza "vivo" e di disponibilità di risorse.

4.2.2 Conoscenza Distribuita e Condivisa

E' possibile estendere il concetto di Augustus 2.0 prendendo in esame un ulteriore aspetto già sottolineato da Galanti, ovvero la necessità di avere a disposizione non tanto l'informazione (ad es. "il censimento") quanto "l'indirizzo" ove reperire ed attivare l'informazione in una rete di conoscenza distribuita in cui l'utilizzatore non deve più occuparsi di conoscere direttamente, ma può appoggiarsi ad altri soggetti.

Il concetto di conoscenza distribuita risulta facilmente applicabile, ad esempio, nel caso della gestione delle risorse e della loro disponibilità: l'utilizzatore non deve più

preoccuparsi di ottenere la disponibilità della singola risorsa, è il fornitore ed il custode dell'informazione che si occuperà di tenere aggiornato il dato. O ancora, l'utilizzatore non deve più controllare continuamente il valore di un sensore, è il fornitore dell'informazione (il sensore in questo caso) che si occupa di notificare all'utilizzatore la modifica del valore registrato secondo alcune regole precedentemente stabilite (es. l'uscita da un determinato range di valori).

Per riportare il concetto ad un esempio legato al mondo del 2.0 possiamo pensare alla gestione di un Wiki: l'utilizzatore non deve occuparsi più direttamente del reperimento dell'informazione fisica; una volta trovata la pagina di interesse gli sarà sufficiente ritornarvi successivamente per trovarvi una informazione aggiornata e puntuale (a patto che il fornitore di tale informazione si preoccupi dell'aggiornamento) senza dover entrare mai in contatto con il fornitore. Nei sistemi più avanzati, l'utilizzatore potrà registrarsi presso il fornitore dell'informazione richiedendo di ricevere notifica di eventuali modifiche, senza più preoccuparsi di controllare periodicamente l'informazione di interesse.

4.2.3 Flessibilità

La metodologia Augustus prevede la definizione di una struttura di coordinamento semplice e flessibile. La flessibilità è intesa come capacità del sistema di adattarsi ad ogni situazione operativa in quanto, come scrive lo stesso Galanti, "l'evento – per quanto previsto sulla carta – al suo 'esplodere' è sempre diverso" ed il sistema deve essere in grado di modificare se stesso sulla base delle necessità contingenti in quel preciso istante.

Una forte flessibilità organizzativa è richiesta, ad esempio, nel caso dei già citati C.O.M (Centro Operativo Misto) in quanto questi centri, per definizione, vengono creati al verificarsi dell'evento, in un luogo baricentrico all'evento stesso e quindi senza alcuna capacità di previsione di creazione di infrastruttura. Questa necessità impone sia una accurata scelta delle architetture hardware da metter in campo (argomento che esula dallo scopo di questo documento) sia la scelta di un'architettura informativa non

predefinita a priori e che sia disponibile ad essere modificata in ogni momento nella sua topologia.

4.3 Architettura: Requisiti Architeturali

Dal punto di vista architeturale, per rendere nella realtà i concetti finora espressi, la soluzione proposta per realizzare una infrastruttura a supporto di Augustus 2.0 si dovrà basare su un'architettura distribuita la quale può si trarre giovamento da uno o più punti centrali che svolgano il ruolo di concentratori e di coordinamento, ma facendo attenzione che la mancanza di tali punti non la renda inutilizzabile. All'interno dell'architettura definiamo *Nodi* tutti i punti dell'architettura stessa, ognuno secondo le proprie funzioni. I nodi non devono essere interconnessi con tutti gli altri nodi dell'architettura, ma devono essere in grado di ottenere le informazioni da qualunque altro nodo collegato all'architettura. In tal senso si ipotizza una struttura simile alla struttura della rete globale internet la quale, pur avendo nei fatti alcuni nodi "principali", permette a qualunque utilizzatore abilitato agli standard di comunicazione, di ricevere o inviare informazioni a qualunque altro nodo della rete. Nodi possono quindi essere Sale di Controllo, C.O.M creati ad hoc per l'emergenza, device operativi per questa o quella organizzazione di soccorso. Unico requisito è la capacità del nodo di comunicare secondo gli standard della architettura stessa.

La definizione di un'architettura distribuita e flessibile che faciliti l'interoperabilità tra diversi nodi permetterebbe, infatti, un supporto IT al progetto di gestione delle emergenze più efficiente ma soprattutto più efficace.

- L'informazione potrebbe "viaggiare" tra un nodo e l'altro senza l'intervento umano.
- L'informazione sarebbe codificata attraverso schemi strutturati e condivisi al fine di renderla inequivocabile e interpretabile da tutti i nodi aderenti all'architettura.

- L'aggiornamento delle informazioni all'interno dell'architettura gioverebbe degli strumenti IT messi a disposizione dall'architettura (primo fra tutti la gestione di eventi).
- Il processo di collaborazione potrebbe trovare nell'architettura un valido elemento abilitante.
- Il concetto di scenario strutturato, definito in seguito, renderebbe possibile la condivisione di interi "Scenari di Rischio" e non solo di singole informazioni, con conseguenti benefici sia in fase di preparazione che al verificarsi dell'evento.

Si descrivono ora in maggior dettaglio alcuni concetti fondanti la nuova architettura sulla base dei quali si intende tendere all'implementazione nella realtà di quanto presentato come Metodologia Augustus 2.0: l'informazione come servizio, la gestione di eventi e soprattutto il concetto di scenario come servizio di servizi

4.3.1 L'informazione come Servizio

Come descritto in precedenza l'utilizzo di architetture SOA rappresenta una risorsa per l'interoperabilità di EMIS. L'utilizzo dei servizi, nella fattispecie servizi web, permette l'astrazione del dato o della funzionalità disaccoppiandola dall'implementazione operativa e rendendo possibile la gestione di informazioni diverse per dominio, fonte e natura. Ogni risorsa, ogni informazione, ogni capacità elaborativa dell'architettura deve avere una propria interfaccia come servizio web permettendone così l'accesso da parte degli altri attori presenti all'interno dell'architettura.

Architettura per la Condivisione delle Informazioni

Il ricondurre l'informazione ad un'astrazione di un servizio semplifica il processo di condivisione delle informazioni stesse potendo sfruttare la natura interoperabile dei servizi stessi. Gli standard operativi basati su SOA già da qualche anno permettono la condivisione di servizi web, fermo restando le necessità e le peculiarità dettate dalla regolamentazione dell'accesso ai servizi (Authorization e Authentication) e della loro disponibilità.

La condivisione delle informazioni, se da un punto di vista architetturale può essere risolta attraverso l'utilizzo di SOA, dal punto di vista dei dati occorre che sia supportata da uno o più standard operativi che permettano la codifica e la decodifica dell'informazione stessa indipendentemente dalla piattaforma applicativa. Il connubio tra architettura di distribuzione, semantica e sintassi dell'informazione garantisce la possibilità di condividere l'informazione stessa anche tra sistemi informativi eterogenei.

Se per quanto riguarda la sintassi dell'informazione, da alcuni anni il meta-linguaggio XML è divenuto lo standard di fatto nello scambio di dati tra sistemi eterogenei, la definizione di singoli linguaggi per lo scambio dei dati dipende dal dominio applicativo di cui l'informazione fa parte.

Nel caso della gestione del rischio, nel capitolo precedente si è analizzato lo stato dell'arte attraverso la presentazione di alcuni standard o proposte di standard relativi alla gestione di eventi calamitosi, sia nella fase di preparazione che in quella di effettivo verificarsi dell'evento. La scelta dei linguaggi da utilizzare è sicuramente uno dei fattori determinanti per la riuscita e la longevità dell'architettura di condivisione delle informazioni, ma sarebbe un errore di progettazione immaginare una soluzione legata a questo o a quello standard in quanto, come già descritto, gli standard nel campo del Risk Management sono molto giovani e quindi facilmente suscettibili a modifiche o cambiamenti.

Per questa semplice osservazione occorre che l'architettura di distribuzione dei contenuti sia fortemente disaccoppiata dalla sintassi e dal linguaggio usato per lo scambio di informazioni in quanto questi risultano forse, l'elemento più soggetto al cambiamento dell'intero progetto. Non bisogna escludere infatti che la stessa architettura di distribuzione possa essere utilizzata per distribuire uno o più standard differenti in modo trasparente, o che alcuni standard stessi subiscano migliorie o vengano addirittura abbandonati per dare spazio ad una codifica dell'informazione migliore dal punto di vista del dominio applicativo.

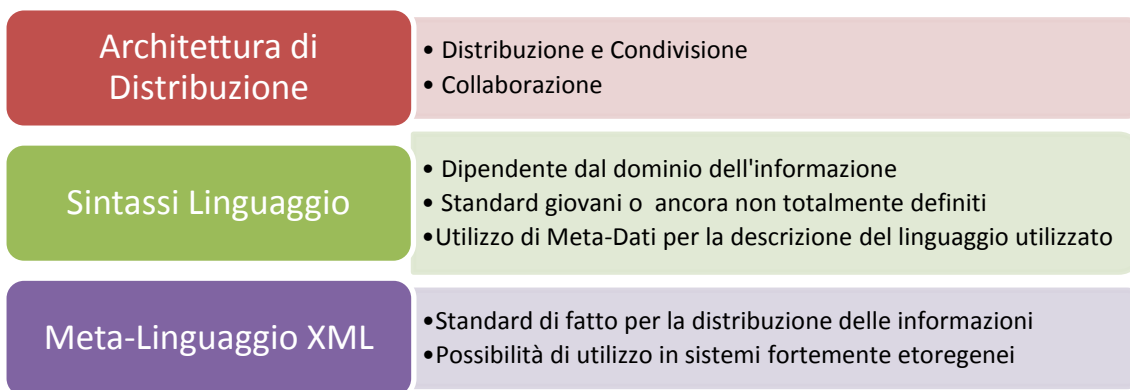


Figura 5 - Architettura e Linguaggi. Peculiarità

Nella scelta e definizione dell'architettura occorrerà quindi tenere conto di questo aspetto fondamentale definendola in modo indipendente dalle informazioni che distribuisce tramite l'utilizzo di meta dati o di descrittori del tipo di informazione trasportata (Figura 5).

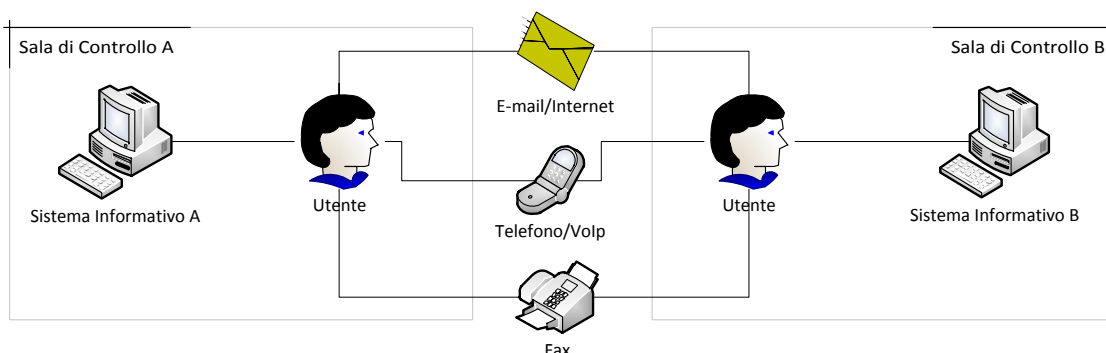
Così come nel caso dei servizi web, la condivisione di un grande numero di informazioni necessita della presenza di un ente che permetta la ricerca delle informazioni all'interno dell'architettura a seconda dei requisiti e delle necessità dell'utente. Tale ente, astratto anche esso in un servizio, che funge da catalogo deve, da un lato, permettere ai fornitori delle informazioni di registrarsi segnalando la loro esistenza e la loro disponibilità a fornire quanto il loro possesso, e dall'altro, permettere ad eventuali utilizzatori di effettuare ricerche per poter accedere e consumare l'informazione stessa. La funzionalità di catalogo risulta essere un punto abilitante e fondamentale per garantire la condivisione e l'utilizzo di informazioni distribuite all'interno dell'intera architettura. Alcuni aspetti rilevanti del servizio di catalogo saranno discusso nei prossimi paragrafi.

Aggiornare le Informazioni tramite Eventi

Dall'analisi svolta sullo stato dell'arte risulta chiaro come il processo di coordinamento delle diverse strutture coinvolte in un piano di emergenza sia fortemente legato alla disponibilità ed all'intervento di personale umano. Benché questo fatto non rappresenti di per sé un problema, occorre evidenziare come il semplice

aggiornamento di alcune informazioni richieda l'intervento dell'operatore in quanto le tecnologie IT a supporto del fornitore e dell'utilizzatore dell'informazione non sono in grado di comunicare in modo automatico. Come già anticipava Galanti nel suo documento di presentazione del metodo Augustus, le risorse e le tecnologie spesso non mancano, manca invece un processo di coordinamento che porti alla creazione di sinergie tra le diverse infrastrutture IT.

La situazione odierna presenta infatti una forte disponibilità tecnologica dei centri di controllo (dalle e-mail alle tecnologie VoIP da webcam e immagini in tempo reale a sensor-network) ma una scarsa integrazione tra centri di controllo o strutture adiacenti. L'informazione c'è, è codificata in un sistema informativo, ma per poterla passare occorre lato fornitore un operatore che digiti una mail e lato utilizzatore un altro che la reinterpreti per poter aggiornare il proprio sistema informativo.



Alla luce di quanto descritto nel paragrafo precedente, risulta chiaro come la definizione di un'architettura per la condivisione delle informazioni possa essere un grande facilitatore nel processo di scambio delle informazioni. L'informazione, astratta come servizio, potrebbe passare da un soggetto all'altro dell'architettura senza intervento umano.

Ipotizzata la presenza di questo facilitatore (l'architettura di scambio delle informazioni), occorre definire le modalità attraverso le quali si desidera mantenere aggiornata l'informazione ma soprattutto "mantenere aggiornati chi l'informazione la utilizza". Detenere informazioni vecchie, non più aderenti alla realtà, può portare ad

errori più gravi del semplice fatto di non averle. È facile immaginare come, in un'architettura per la condivisione all'interno di un EMIS, il numero di informazioni presenti sia notevole e quali difficoltà possano emergere nella gestione di un sistema efficace di aggiornamento alla luce anche del grande numero di utilizzatori diversi, impossibile da stabilire a priori.

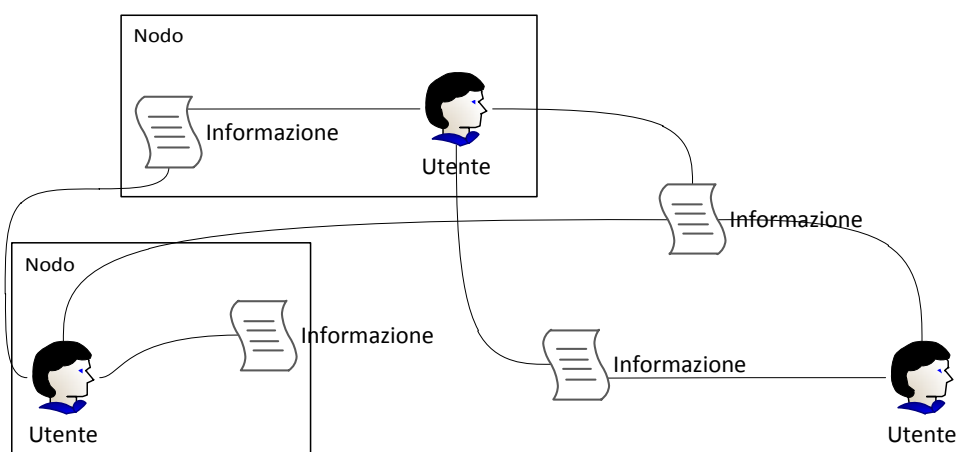


Figura 6 - Topologia Rete Informazione Utilizzatore

Per rispondere al meglio all'esigenza di fornire sempre informazioni aggiornate e congruenti allo stato reale, è possibile ipotizzare l'estensione dell'architettura finora definita attraverso l'introduzione del concetto di evento, facendo quindi coesistere sia la logica di comunicazione push che pull.

A tal proposito, è stato già presentata, nel corso del capitolo la peculiarità dei Wiki e del loro sistema di aggiornamento. Spesso nelle applicazioni 2.0 al verificarsi di un aggiornamento di un contenuto all'interno di un Blog o di un Wiki, l'applicazione notifica tale modifica agli utenti "iscritti" al servizio tramite mail o RSS.

Tale notifica basata sull'evento (in questo caso di aggiornamento) risulta molto utile in ottica di gestione della disponibilità di risorse o nel monitoraggio di particolari dati tipica di EMIS. All'interno dell'architettura in ottica Augustus 2.0 occorre quindi predisporre una metodologia per la gestione di eventi, la quale definisca i processi di registrazione e di notifica legati all'aggiornamento di informazioni in modo trasparente

alle modalità con cui tale aggiornamento è avvenuto, ad esempio tramite intervento umano o procedura automatica.

L'invio di una notifica al verificarsi di un dato evento (modifica di un valore, valore uscito da un determinato range, servizio non più disponibile o altro) risulta uno strumento utile se non necessario nel perseguire, grazie al supporto informativo, l'obiettivo di Piani di Emergenza o scenari di Rischio attuali, quasi in real-time. L'innalzarsi del livello di un lago oltre un certo livello, la modifica del numero di addetti all'interno di una grande industria, lo spostamento di una caserma dei carabinieri, la dismissione di alcuni mezzi dei vigili del fuoco sono solo alcuni degli eventi che, una volta notificati, possono indurre all'aggiornamento dello scenario o addirittura alla sua ristesura.

L'idea di fondo resta quella di portare la responsabilità dell'aggiornamento dell'informazione dall'utilizzatore (che poco sa della natura e delle logiche che stanno dietro all'informazione) al fornitore che è invece l'unico responsabile della veridicità e correttezza dell'informazione. Grazie a questo ribaltamento di responsabilità, sempre controllato dall'utilizzatore che può richiedere o meno la registrazione al particolare evento, l'aggiornamento risulta più tempestivo e flessibile, ma soprattutto più corretto dal punto di vista formale e quindi meno soggetto ad errori o dimenticanze. Ricordando infine l'astrazione proposta "informazione come servizio" risulta chiaro come sia compito dell'interfacciamento tra SI interno e nodo dell'architettura controllare ed "accorgersi" dell'evento per poi ribaltarlo secondo le procedure offerte dall'architettura stessa.

Ricerca le informazioni in Ambiente Distribuito

L'architettura fin qui descritta cerca di rispondere alle esigenze di condivisione delle informazioni e collaborazione, strutturandosi secondo un modello topologico distribuito capace di adattarsi alle diverse situazioni operative attraverso il concetto di "informazione come servizio" e di "notifica di eventi". In un ambiente così complesso e

difficilmente definito a priori emerge la necessità di fornire un valido strumento di supporto alla ricerca, pubblicazione e catalogazione delle informazioni.

La domanda a cui occorre rispondere è la seguente: come può il nodo A reperire un'informazione presente in architettura e detenuta dal nodo D non conoscendo neppure l'esistenza del nodo D? Nel paragrafo precedente abbiamo definito come un nodo può reperire un'informazione (fatti salvi gli aspetti autorizzativi) da qualunque altro nodo presente nell'architettura. Tale comportamento richiede l'esistenza di un catalogo, come già accennato nel capitolo 3, ove i diversi nodi possano pubblicare le informazioni in loro possesso (sotto forma di servizi), catalogo che, per quanto affermato precedentemente, non è stabilito essere un nodo centrale a cui tutti i nodi periferici sono collegati.

Il servizio stesso di catalogo deve poter essere distribuito e deve poter risiedere su uno o più nodi, garantendo così le funzionalità di ricerca delle informazioni sopra descritto.

Di seguito si definisce un breve esempio di utilizzo di un catalogo distribuito il cui approfondimento esula dallo scopo di questo documento, con il solo scopo di fornire una visione d'insieme sull'architettura a supporto dell'idea Augustus 2.0.

Come accennato, il nodo A necessita di un informazione che è detenuta dal nodo D: il nodo A inoltrerà la richiesta di informazione all'unico nodo con servizio di catalogo a cui è collegato (nodo B) il quale dovrà fornire come risposta l'informazione o quantomeno il luogo¹ ove reperirla. A seguito della richiesta del nodo A, sono due gli scenari che si aprono:

- Il nodo B conosce ove reperire l'informazione (ovvero l'informazione risiede sul nodo B stesso o su un altro nodo collegato a B): in tal caso il nodo B fornirà al nodo A il luogo o l'indirizzo ove reperire l'informazione
- Il nodo B non conosce ove reperire l'informazione: in tal caso l'architettura deve permettere la richiesta in cascata da parte di B ai nodi a cui B è collegato

¹ Per luogo si intende il nodo sul quale risiede il servizio astrazione dell'informazione

dotati di catalogo. Il nodo B “prende in carico” la richiesta e si occupa esso stesso di reperire il luogo o l’indirizzo dell’informazione.

Una volta ottenuta l’informazione richiesta il Nodo B può fornire al Nodo A l’indirizzo dell’informazione stessa, permettendone il reperimento del dato.

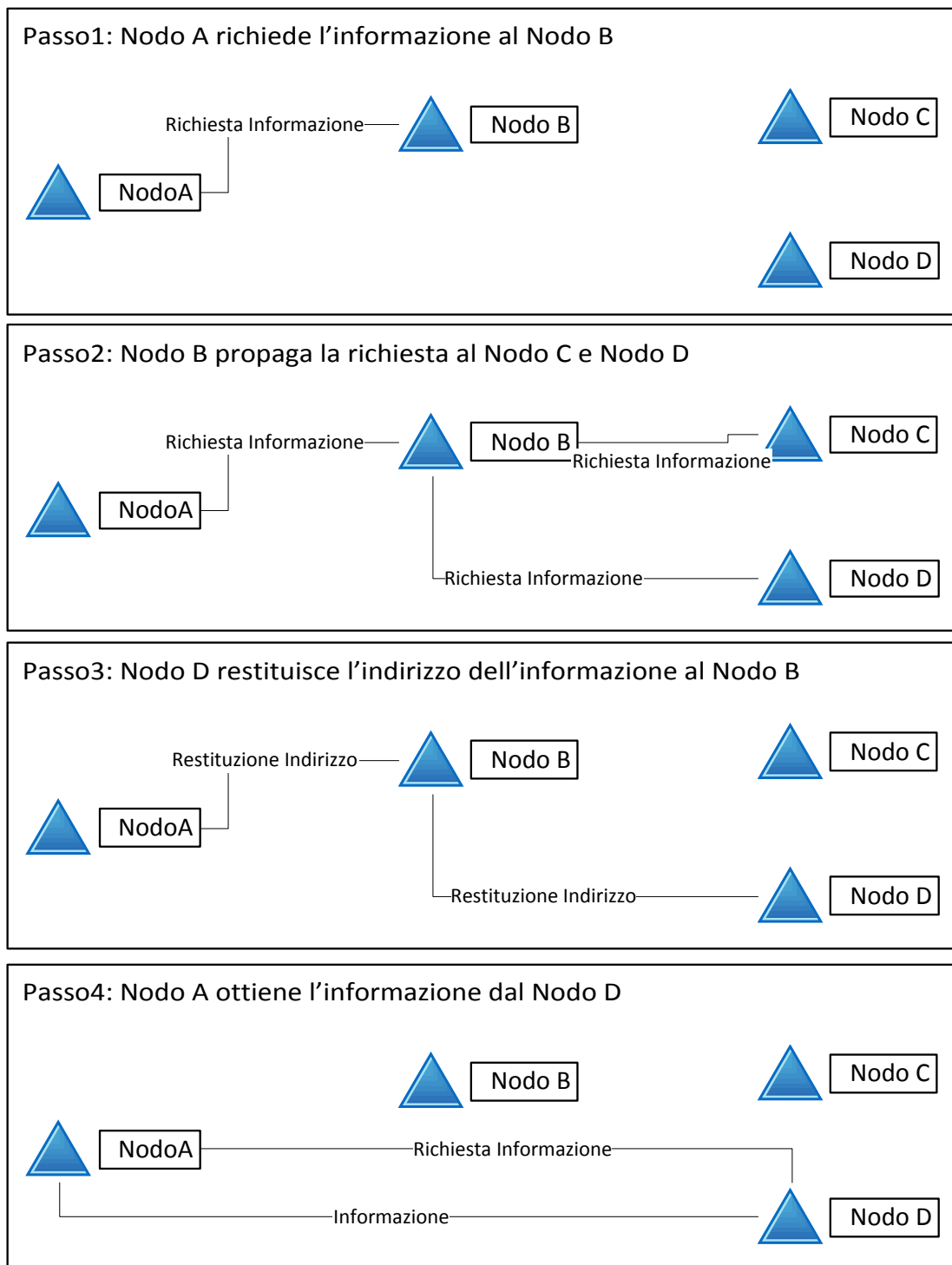


Figura 7 - Ricerca in cascata in ambiente distribuito

4.3.2 Lo Scenario come Servizio di Servizi

Uno dei punti fondamentali della metodologia Augustus è la definizione di scenari di rischio e il loro continuo aggiornamento; sfruttando il supporto del sistema informativo proposto ed analizzato, si configura una nuova metodologia a supporto della creazione e manutenzione di Scenario di Rischio. Ad una prima fase di definizione dello scenario attraverso la raccolta dei dati disponibili all'interno dell'architettura ne segue una seconda, trasparente all'utente di continuo aggiornamento del piano grazie alla notifica degli eventi di modifica.

Tale aggiornamento può, se ritenuto utile, scatenare allerte sullo scenario stesso, sia dovute a particolari eventi naturali in atto e monitorati da sensori real-time sia dovute alla modifica delle risorse disponibili o a cambiamenti ritenuti "importanti" all'atto della definizione dello scenario stesso. Lo scenario passa quindi da un aggiornamento batch eseguito dall'utilizzatore delle diverse informazioni e creatore dello scenario ad un aggiornamento continuo dato dalla collaborazione tra fornitore dell'informazione e scenario stesso.

In definitiva ciò che si viene a creare è l'unione di oggetti differenti rappresentanti le diverse informazioni ritenute utili in sede di design dello scenario "appoggiati" sulla rappresentazione virtuale di un dato territorio. Lo scenario risulta, in senso astratto, essere la composizione di diverse informazioni e quindi di diversi servizi presenti all'interno dell'architettura, ponendo le basi per alcune funzionalità a supporto delle decisioni in ambiente EMIS.

Il servizio scenario

Lo scenario così definito può diventare esso stesso un servizio disponibile ed utilizzabile dagli altri nodi all'interno dell'architettura, da condividere e distribuire secondo le direttive dell'architettura. L'aggiunta del nuovo servizio può essere quindi notificata al Catalogo di riferimento per essere esposto e pubblicato secondo le procedure previste. La chiamata al servizio di Scenario comporta la restituzione al Nodo chiamante dell'intero insieme delle informazioni, dello stato e delle risorse

utilizzate in quel preciso istante oltre alla possibilità di ottenere gli indirizzi ove reperire le singole informazioni.

Lo scenario stesso diventa quindi un'informazione utilizzabile per l'elaborazione di Piani di Emergenza, per la gestione di nuovi Scenari o per la modifica dello stesso su un altro Nodo. Lo scenario può implementare la notifica di eventi dovuti ad una sua modifica complessiva, o alla modifica di alcuni valori in esso contenuti, mentre per ottenere una gestione di eventi a livello di granularità della singola informazione, sarà compito del nuovo Nodo registrarsi presso i singoli indirizzi forniti per ottenere le notifiche di evento.

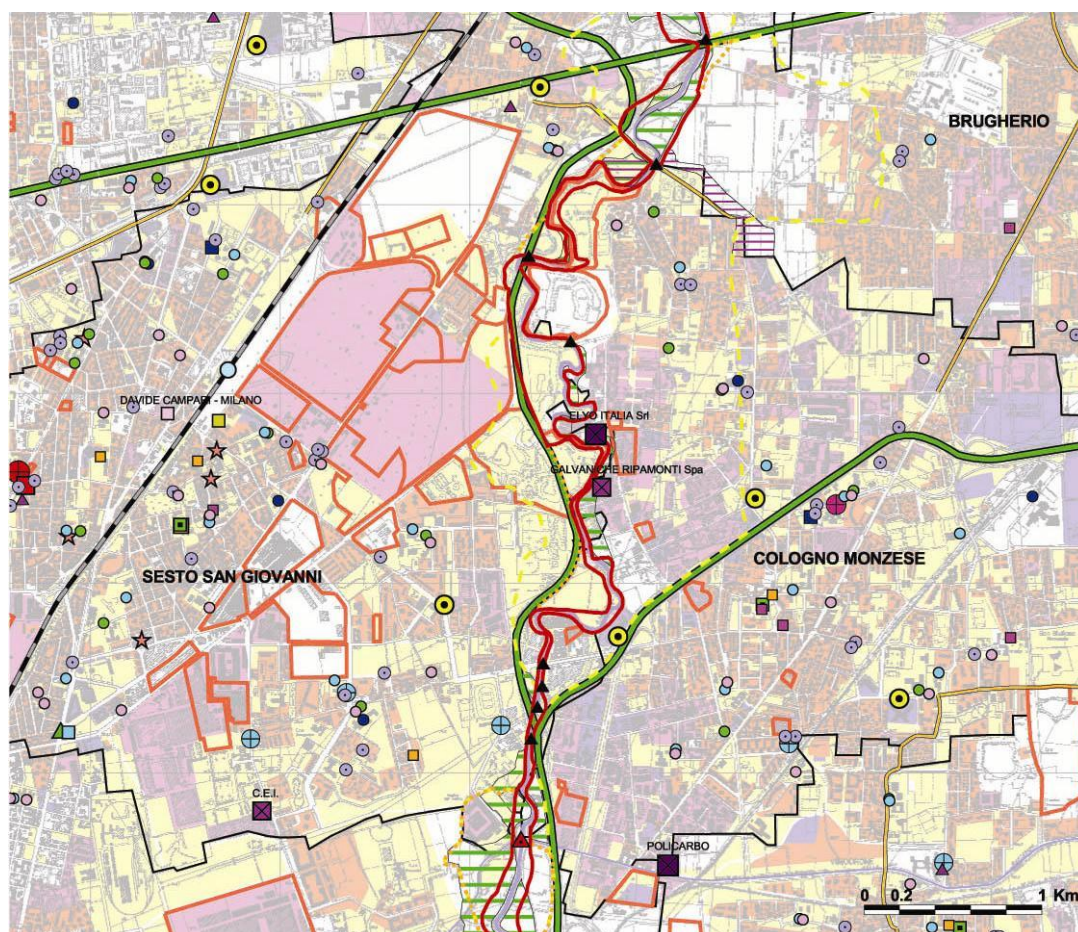


Figura 8 - Esempio di Scenario

Database Temporale

Allo scenario può essere agganciata una base dati temporale che possa immagazzinare lo stato dello scenario stesso sotto forma screenshot² delle informazioni presenti, dello stato delle risorse e della loro collocazione sul territorio. Il database temporale potrebbe quindi contenere le fotografie dello scenario costantemente aggiornato attraverso la notifica di eventi. Il salvataggio dello stato dello scenario rende possibile inoltre la gestione tramite sistema informativo della storicità dell'area e del rischio ad esso collegato attraverso la creazione di apposite funzionalità per la ricerca di dati storici.

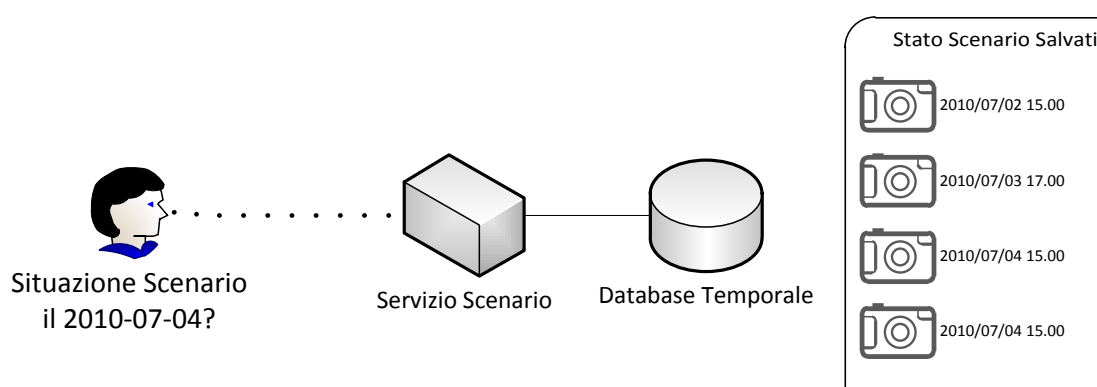


Figura 9 - Utilizzo del Database Temporale

Evoluzione basata su dati Storici

Lo scenario come servizio permette la distribuzione ad altri nodi della situazione in tempo reale così come la possibilità di recuperare un particolare stato o screenshot dello scenario a patto che lo scenario implementi il supporto al DB temporale. La richiesta di screenshot passati permette la visualizzazione ed il confronto analitico tra due situazioni verificatesi nel corso del tempo nella stessa area, in aree adiacenti o in aree ritenute dall'intervento umano simili nella composizione o nei comportamenti nel dominio dell'evento in analisi.

² Con il termine Screenshot si identifica la fotografia dello scenario in un preciso istante come somma di dati strutturati (Mappa + Informazioni)

L'utilizzo del DB temporale permette inoltre la ricerca, dato uno scenario, di situazioni passate simili ad una data situazione attuale attraverso una ricerca per similarità date alcune informazioni, stati o risorse dello scenario. Ad esempio, dato uno scenario con forte rischio idrogeologico, l'operatore umano, conoscitore del dominio applicativo e naturale dell'evento, può richiedere al servizio la ricerca di screenshot simili a partire dai dati di livello idrometrico del fiume in un dato punto e di previsione di 100mm/18 h di pioggia. Il servizio restituirà all'operatore le situazioni che rispondono ai parametri di ricerca e gli permetterà la visualizzazione dell'evoluzione che a partire da tale screenshot passato lo scenario ha vissuto. Tale funzionalità risulta utile nella programmazione di eventi ciclici o nell'analisi a posteriori dell'accaduto.

Lo scenario all'interno di Augustus 2.0

Riassumendo i concetti enunciati nei paragrafi precedenti:

- L'utente può creare uno scenario come aggregazione di informazioni attraverso la ricerca all'interno dell'architettura
- L'utente può registrare lo scenario presso i diversi servizi di informazione al fine di ottenere dati aggiornati sulla base di query predefinite.
- L'utente può pubblicare lo scenario all'interno dell'architettura creando così una nuova fonte di informazione
- L'utente può definire uno scenario abilitato alla gestione degli eventi al fine di notificare eventuali variazioni dello scenario agli utenti che si registrassero
- Dato uno scenario pubblicato, un generico utente può richiedere la visualizzazione dello stesso. Lo scenario restituisce i dati presenti in quel preciso istante, al pari di una fotografia
- Dato uno scenario pubblicato, un generico utente può richiedere la struttura (o configurazione) dello scenario al fine di ricrearlo sul proprio client al fine di modifica o di ulteriore analisi. In tal caso lo scenario restituisce non le informazioni, quanto gli indirizzi ove reperirle. Qualora l'utente desideri ottenere le informazioni o registrarsi per la notifica di eventi dovrà interrogare i diversi servizi presenti nella struttura.

- Dato uno scenario pubblicato è possibile agganciare una base dati temporale per il salvataggio dei diversi stati registrati nel tempo
- Dato un scenario pubblicato e abilitato al salvataggio di dati temporali, è possibile effettuare query basate su date o similarità di informazioni contenute
- Dato un scenario pubblicato e abilitato al salvataggio di dati temporali è possibile visualizzare l'evoluzione a partire da un dato istante di tempo

Capitolo 5

Principi Architettureali

In questo capitolo si intende descrivere una possibile definizione architettureale dei concetti espressi al capitolo 4 quali la gestione di eventi in Risk Management Information System e il concetto di Scenario come servizio di servizi sfruttando l'architettura Orchestra.

5.1 L'architettura Orchestra

La ricerca relativa a possibili soluzioni architettureali esistenti ha individuato nell'architettura descritta nel progetto ORCHESTRA un valido punto di partenza per la progettazione ed implementazione di un sistema di gestione del rischio che integri le soluzioni proposte.

La forza di ORCHESTRA sta nella sua completa specificità in forma astratta, utilizzando il linguaggio UML, al fine di operare una netta separazione tra tecniche implementative, sempre in evoluzione e facilmente suscettibili a cambiamenti ed evoluzioni, e concetti architettureali.

L'architettura orientata ai servizi descritta in ORCHESTRA, l'OA (Orchestra Architecture), è completamente aperta, distribuita e interoperabile: per accedere ad un servizio esposto nell' OSN, l'Orchestra Service Network, occorre solamente implementare le interfacce di accesso e fruizione definite nell' OA; operando in tal

modo sistemi già esistenti possono dialogare e cooperare superando limitazioni amministrative, tecnologiche e linguistiche.

In seguito verranno messi in luce le componenti core dell'architettura ponendo l'attenzione sui punti di forza e debolezza del sistema integrando una possibile estensione dei principi descritti al fine di ottenere il completamento dell'architettura introducendo i concetti e gli strumenti necessari a descrivere la nuova architettura dei sistemi di gestione del rischio.

5.1.1 Domini Funzionali dell'ORCHESTRA Service Network

L'architettura Orchestra deve far fronte al problema di integrare i diversi sistemi di gestione del rischio che sono collegati nelle e tra le organizzazioni. È l'*Orchestra Service Network* (OSN), in veste di istanza in esecuzione di una architettura Orchestra, che contribuisce ad aumentare la interoperabilità semantica e sintattica di questi sistemi.

Le componenti di un OSN, come ad esempio le Orchestra Service Instance (OSI), sono classificate in accordo con i seguenti domini funzionali:

User Domain: fornisce l'interfaccia utente (umano o software) e interagisce con i servizi facente parte del *Mediation and Processing Domain* in accordo alle regole descritto nel Meta Modello di Orchestra. Tuttavia i componenti utente possono interagire tra di loro in modo nativo in quanto parte di una applicazione. Esempi di componenti che fanno parte di questo dominio sono ad esempio i GIS, Sistemi Informativi di enti come la croce rossa, dispositivi mobili o semplicemente applicazioni stand-alone per la consultazione dei dati.

Mediation and Processing Domain: in questo dominio sono presenti le principali componenti funzionali dell'intera architettura. Esso effettua in primo luogo una mediazione delle richieste di servizio inviate dallo *User Domain* verso l'*Integration Domain* e tale mediazione viene eseguita in base alle meta informazioni scambiate con i componenti dell'*Integration Domain*.

In secondo luogo è possibile che le componenti interne del dominio interagiscono tra di loro in accordo con le regole dettate dal meta modello. In questo dominio esistono Istanze di servizio dedicate all'aspetto semantico che cercano di risolvere le differenze semantiche dei modelli di informazione utilizzati dai sistemi sorgenti attraverso l'uso di ontologie. Quindi il *Mediation and Processing Domain* rende possibile interoperabilità semantica, se richiesto dai componenti dello *User Domain*.

Integration Domain: fornisce il supporto per l'integrazione per i sistemi sorgente. Le istanze di servizio in questo dominio hanno due tipi di interfacce: Da un lato esse interagiscono in accordo alle regole dettate dal Meta Modello con i servizi presenti nel dominio di Mediazione; dall'altro esse dialogano con le componenti del *Source System Domain* attraverso le loro interfacce native. Questo dominio quindi rende possibile l'interoperabilità sintattica in un OSN.

Source System Domain: incorpora i sistemi e i componenti di sistema di applicazioni tematiche (ad esempio applicazioni per la gestione del rischio) che devono essere integrati nel Network. Essi sono la sorgente dei dati e delle funzionalità e sono collegati con l'*Integration Domain*. Per far ciò occorre effettuare uno sforzo di re-ingegnerizzazione dei sistemi per poter mappare le funzionalità e le modalità di accesso ai dati con una interfaccia conforme ad un servizio Orchestra.

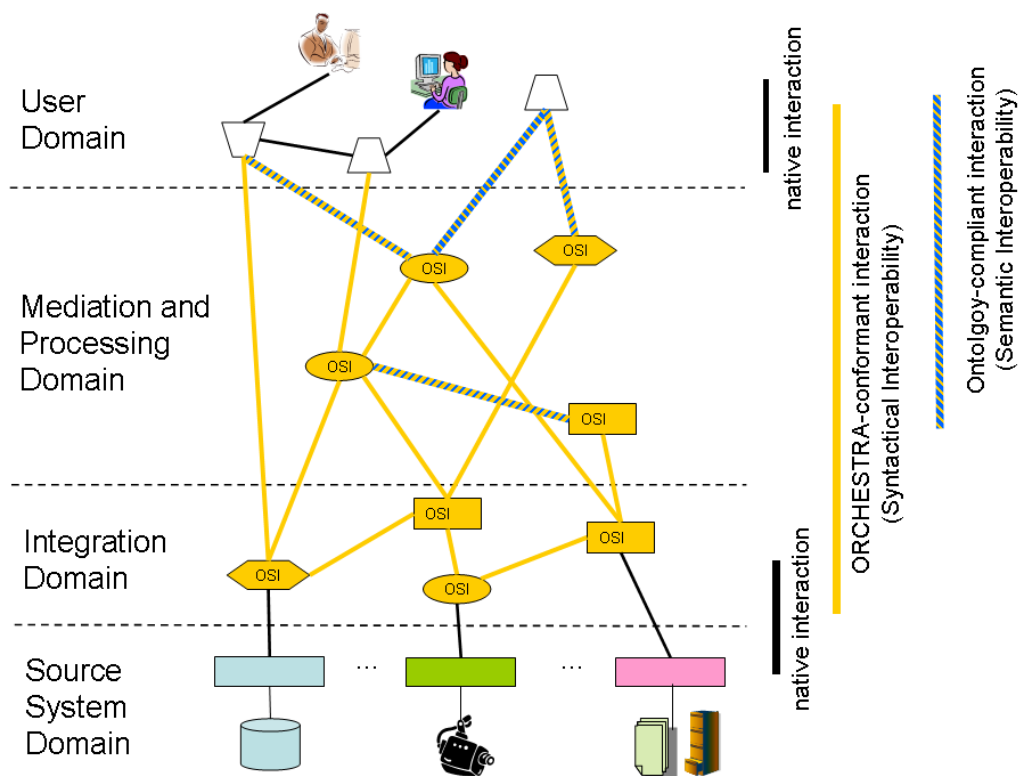


Figura 10 - Domini Funzionali in un Orchestra Service Network
(RM-OA Rev2.1 as OGC Best-practice document 07-097)

5.1.2 L'interoperabilità semantica: Information Meta-model e Service Meta-model

L'architettura Orchestra è generica per definizione in quanto si astrae dalla specifica implementazione e non prescrive l'uso di uno specifico modello di informazione o di configurazione delle istanze dei servizi relativi a un determinato contesto applicativo. L'architettura fornisce invece un set di elementi di base che devono essere usati e un set di regole che devono essere osservate al fine di garantire l'interoperabilità sintattica e semantica tra le componenti software di una applicazione Orchestra. Queste elementi base e regole sono descritte nell' Orchestra Meta-model (OMM).

L' OMM si compone di due parti:

L'*Orchestra Information Meta-Model* (OMM-Information) fornisce le regole sulle modalità di specifica dei modelli di informazione e dei modelli di meta-informazione e prescrive l'uso dei tipi di dato.

L'*Orchestra Service Meta-model* (OMM-Service) descrive le regole di specifica delle interfacce e dei servizi Orchestra e propone una serie di servizi architetturali da usare in un Orchestra Service Network.

Entrambe le parti dell'OMM sono interrelazionate e dipendenti le une dalle altre. Da una parte le strutture dei parametri di input e output delle operazioni dichiarate nelle interfacce devono rispettare le regole dell' OMM-Information. Dall'altra, operazioni pre-costruite sui applicazioni già esistenti devono rispettare le regole imposte dall'OMM-Service.

Information Meta-model

Nel contesto di un Orchestra Service Network i modelli di informazioni sono specificati al fine di regolare la dichiarazione delle strutture dati che permettono il corretto scambio di informazioni tra i diversi servizi presenti nell'OSN. Il ruolo dell'Information Meta-model consiste nel definire le regole per la specifica di questi modelli di informazione (chiamati Orchestra Application Schema OAS) con l'intento di ottenere un approccio armonico per la specifica di tutti i servizi e con questo rendere possibile il riuso e l'interoperabilità tra i componenti software.

Il concetto centrale dell'OMM-Information è la definizione di ***feature*** come unità informativa di base e l'OMM-Information si occupa della specifica dei meta-modelli delle feature.

Una feature è una astrazione di un fenomeno del mondo reale ed è possibile raggruppare diverse istanze di feature in feature type quando esse hanno le stesse caratteristiche. Un feature type contiene un elenco di proprietà che possono essere attributi, operazioni o associazioni con altri feature type. In una logica ad oggetti questi feature type possono essere estesi per ereditarietà per raffinare concetti già esistenti.

L'OMM-Information specifica anche i domini relativi ai valori che possono assumere gli attributi: a esempio specifica l'elenco dei tipi primitivi da cui bisogna partire per definire i meta modelli.

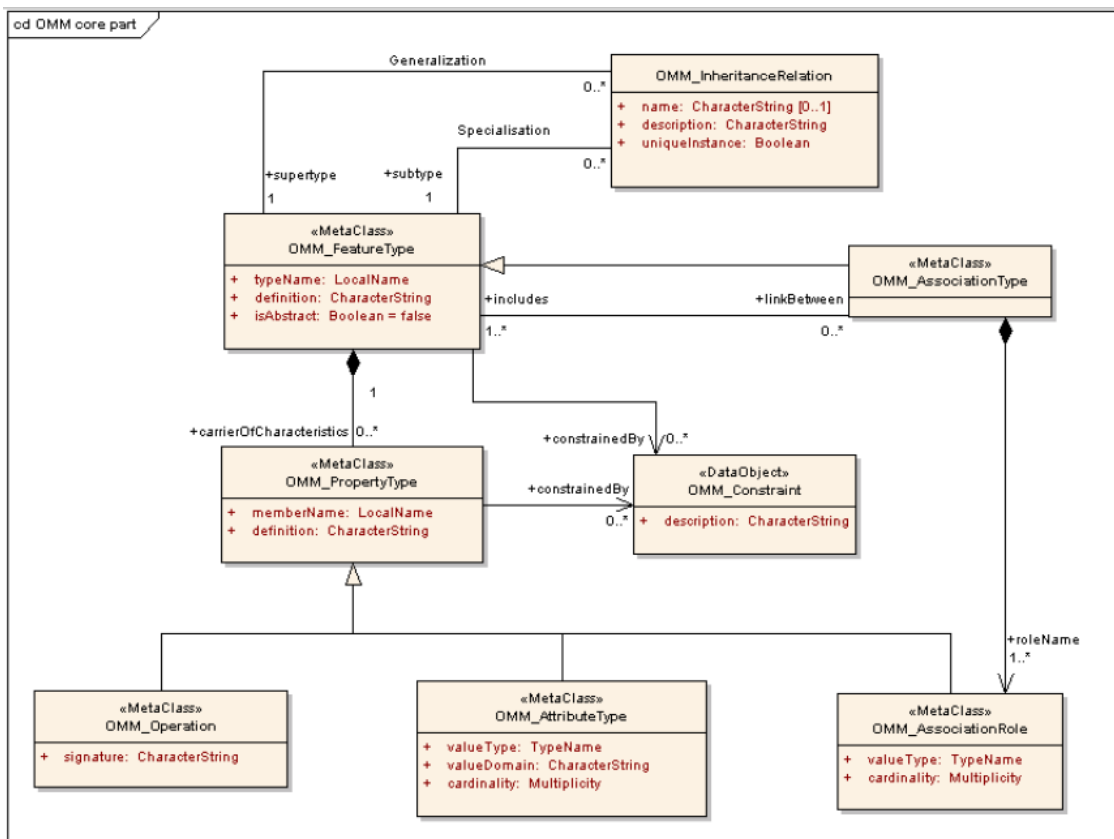


Figura 11 - Parte principale del Orchestra Information Meta-model (RM-OA Rev2.1 as OGC Best-practice document 07-097)

Service Meta-model

Il Service Orchestra meta-model fornisce un meta-modello e delle regole per la specifica di un servizio Orchestra; seguendo la filosofia di fondo del progetto l'approccio alla modellazione e alla specifica del servizio non è legata alla particolare piattaforma sulla quale il servizio sarà installato ma deve rimanere astratta per favorire il riuso delle specifiche dei servizi resi disponibili da diverse applicazioni abilitando così una interoperabilità inter-piattaforma.

OMM-Service definisce un Servizio Orchestra come una serie di interfacce che specificano esternamente il comportamento del servizio. Il concetto di interfaccia cerca di aggregare funzionalità coerenti al fine di raggiungere un particolare obiettivo (ad esempio il rendering di informazioni geografiche in una mappa) che potrà essere riusato nella composizione di altri servizi.

Una operazione è specificata dalla sua signature, come ad esempio il suo nome e i suoi parametri di input e output. Qui il collegamento tra OMM-Service e l'OMM-Information diventa visibile: I tipi dei parametri di richiesta e di risposta sono strutturati in un Orchestra Application Schema (OAS) che segue le regole dell' OMM-Information. Il valore di un parametro può essere perciò un dato primitivo, come un integer, ma anche un set di feature.

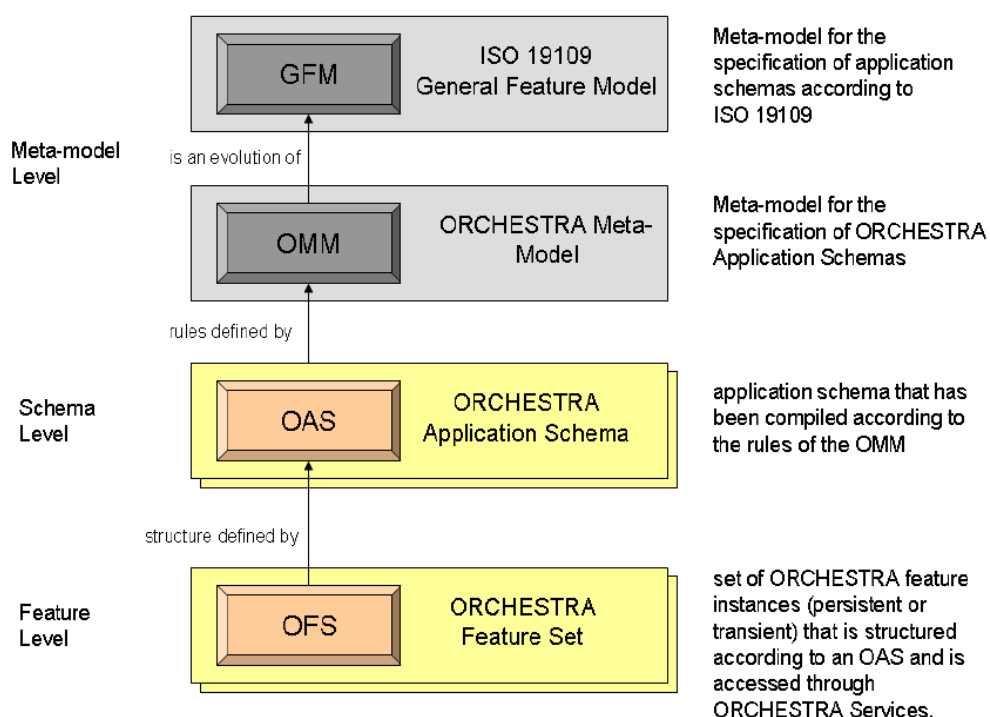


Figura 12 - Modelli di Informazione di Orchestra (RM-OA Rev2.1 as OGC Best-practice document 07-097)

5.1.3 I Nodi Orchestra

Il service network implementato seguendo i principi dell'architettura Orchestra coinvolgerà diversi tipi di componenti come ad esempio un GIS che mette a disposizione un servizio di reperimento delle mappe, un sensore o una rete di sensori che rileva il livello idrometrico di un fiume o un DDS di un ente pubblico che reperisce delle informazioni per poi procedere con una valutazione della situazione del rischio.

Questi componenti quindi possono essere sia fornitori di un servizio (GIS e sensori) o utenti (DDS) e ognuno di essi in base alle specifiche dell'architettura implementerà la

relativa interfaccia di accesso, ricerca e fruizione del servizio; ognuno di questi punti, indipendentemente dalla funzione svolta, è definito un “Nodo Orchestra”.

L’idea di base è che ogni Nodo Orchestra non sia a conoscenza di tutti i nodi e servizi che sono associati al network ma solo di alcuni di questi: solo a fronte di un reale bisogno di informazione verrà messo a conoscenza dell’esistenza di altri Nodi attraverso il servizio di ricerca (vedi Catalogo); esso salverà internamente l’end-point di questo nuovo Nodo e potrà utilizzare i servizi forniti da quest’ultimo utilizzando sempre il modello di interfacce Orchestra.

Se un nuovo Nodo entra a far parte del network esso dovrà necessariamente conoscere almeno uno degli altri nodi; una volta che questo è connesso a tale nodo i suoi servizi saranno resi ricercabili e disponibili a tutti gli altri nodi del network.

Questa architettura implementa il concetto di conoscenza distribuita, ovvero ogni Nodo non deve conoscere effettivamente l’informazione che gli serve ma dove andare a reperirla sapendo che essa è costantemente aggiornata dal fornitore di tale dato.

5.1.4 Il servizio Base Orchestra: OA Basic Service

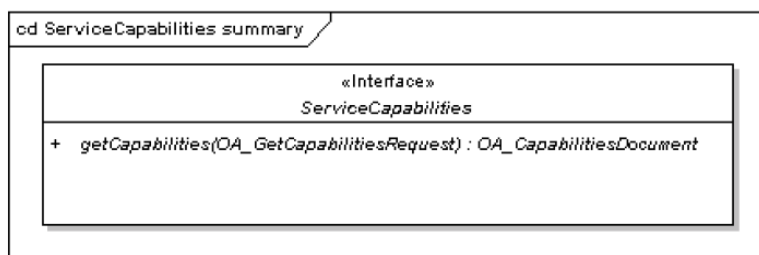
L’OA Basic Service fornisce la descrizione di tutte quelle funzionalità che sono richieste dai servizi che usufruiscono dell’architettura Orchestra. Questa descrizione è fornita attraverso la definizione di una serie di interfacce astratte che possono essere estese o adattate dai vari servizi Orchestra. Inoltre questo servizio definisce alcuni tipi di eccezione predefiniti potenzialmente utilizzabili da tutti i servizi Orchestra.

La definizione delle interfacce di tutti i tipi di servizi Orchestra da parte dell’OA Basic Service fa sì che tale servizio non venga in effetti mai implementato nella pratica poiché è troppo generico: solo alcune delle sue interfacce verranno implementate dai servizi tematici.

Il servizio si compone quindi di diverse interfacce:

Service Capabilities Interface: l'obiettivo di questa interfaccia consiste nel definire una modalità uniforme per poter recuperare una auto descrizione di un servizi attraverso l'utilizzo delle suddette capabilities. Le capabilities forniscono le meta-informazioni relative al servizio a cui sono state richieste.

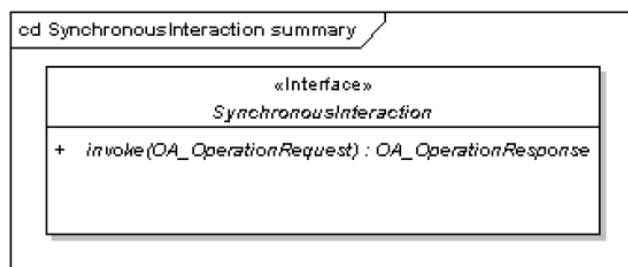
Tutti i servizi Orchestra devono necessariamente implementare questa interfaccia.



*Figura 13 - Interfaccia Service Capabilities
(RM-OA Rev2.1 as OGC Best-practice document 07-097)*

Synchronous Interaction Interface: L'obiettivo di questa interfaccia è la definizione di una modalità uniforme per effettuare delle richieste di esecuzione sincrona di una operazione.

Per esecuzione sincrona di un'operazione si intende che il client dopo aver richiesto l'attivazione dell'operazione attende finché essa non è conclusa e quindi che gli venga ritornato un messaggio di risposta che può contenere il risultato dell'operazione o un'eccezione dovuta a qualche problema durante l'esecuzione della richiesta.



*Figura 14 - Interfaccia Synchronous Interaction
(RM-OA Rev2.1 as OGC Best-practice document 07-097)*

Asynchronous Interaction Interface: in modo duale rispetto all'interfaccia sincrona, l'obiettivo di questa interfaccia consiste nella definizione delle modalità per richiedere l'esecuzione di un'operazione ad un servizio in modo asincrono.

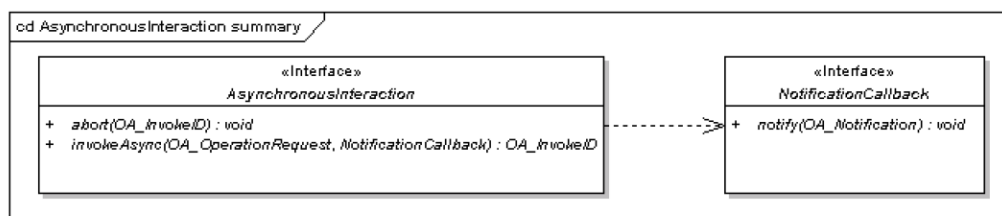


Figura 15 - Interfaccia Asynchronous Interaction
(RM-OA Rev2.1 as OGC Best-practice document 07-097)

5.1.5 Authentication e Authorization

Pur non avendo parte attiva nei concetti proposti, per completezza si riportano le caratteristiche del servizio di autorizzazione ed autenticazione, in una visione più completa di architettura volta all'implementazione della metodologia Augustus 2.0

L'architettura definita nel progetto Orchestra è di tipo aperto ma ciò non significa che meccanismi di verifica della genuinità dei servizi interfacciati al network e l'autenticità degli utenti non siano presenti.

Nell'architettura analizzata anche i sistemi di autenticazione e autorizzazione sono forniti sotto forma di servizi: essi sono l'*Authentication Service* e l'*Authorization Service*.

Questi sono servizi distribuiti che si basano sul concetto di autorizzazione condivisa implementando un sistema di sessione; una volta che il nodo è stato autenticato esso riceve un oggetto che descrive le informazioni sulla sessione da utilizzare per l'accesso ai servizi del network senza dover effettuare nuovamente il login: sarà carico dei nodi verificare la bontà della sessione e delle autorizzazioni concesse prima di fornire il servizio.

Authentication Service

Il servizio di autenticazione verifica la genuinità degli attori utilizzando il set di credenziali fornite. Nella specifica dell'architettura viene descritta solamente l'interfaccia dei metodi di autenticazione; il modo in cui essa viene realmente effettuata è lasciato all'implementazione. Ad esempio il servizio può essere implementato da un'applicazione che interroga un database relazionale centralizzato gestito da un ente supervisore o essere l'unione di più servizi di autenticazione realizzato con tecnologia Web Service della quale però l'utente non è a conoscenza né soprattutto ha interesse di conoscere. Quali credenziali necessita il sistema di autenticazione e il modo con cui sono trasmesse è specifico del sistema di autenticazione. Le informazioni di sessione ritornate dal servizio a fronte di una corretta autenticazione possono essere usate per invocare successivamente altri servizi.

Possiamo suddividere le funzionalità fornite dal servizio in 3 categorie: accesso, controllo e gestione.

Nella categoria accesso troviamo quel metodo che permette l'autenticazione vera e proprio al network:

```
login(OA_Principal, OA_Credentials, OA_AuthenticationSessionInformation)  
: OA_AuthenticationSessionInformation
```

Passando al metodo le credenziali relative ad un utente si riceverà un oggetto di sessione da utilizzare per le successive interazioni con il sistema.

Nella specifica del metodo notiamo le entità *OA_Principal* e *OA_Credential*: *OA_Principal* individua l'attore e *OA_Credential* rappresenta le credenziali che devono essere verificate; ad esempio in un semplice sistema di controllo di username e password testuale la username rappresenta l' *OA_Principal* mentre la password l' *OA_Credential*.

Per controllo si intende la funzionalità di verifica delle informazioni di sessione:

```
verifySessionInformation(OA_AuthenticatedPrincipal) : Boolean
```

Richiamando il metodo *verifySessionInformation* sarà possibile verificare l'effettiva identità del nodo che richiede il servizio da parte del fornitore di quest'ultimo.

Infine nella categoria gestione troviamo tutte quelle funzionalità proposte all'inserimento, modifica, cancellazione di un determinato utente dal network. Alcuni esempi di metodi sono:

```
activatePrincipal(OA_Principal) : void  
addCredentials(OA_Credentials, OA_Principal) : void  
createPrincipal(OA_Principal) : void  
deactivatePrincipal(OA_Principal) : void  
removeCredentials(OA_Principal) : void  
removePrincipal(OA_Principal) : void
```

Authorization Service

Il servizio di Autorizzazione decide se un utente autentificato è autorizzato all'accesso a una determinata risorsa. A fronte di una richiesta di servizio, il fornitore del servizio richiama il servizio di autenticazione in modo da reperire l'esplicita autorizzazione all'accesso. La decisione di autorizzare o meno il nodo che effettua la richiesta si basa sul contesto di autorizzazione contenuto nelle informazioni sulla sessione del nodo passate al metodo le quali sono state reperite in fase di autenticazione.

La risposta del servizio di autenticazione contiene un valore di conformità della richiesta che indica se permettere o rifiutare l'accesso alla risorsa.

Il metodo principale del servizio si occupa dell'autenticazione vera e propria:

```
authorise(OA_AuthorisationContext) : CharacterString
```

Il metodo *authorize* implementa la procedura di autenticazione: durante una richiesta di autenticazione il servizio recupera un *AuthorizationContext* che contiene gli utenti autenticati e le informazioni di esecuzione del servizio in questione. Grazie all'elenco

degli utenti autenticati il servizio di autenticazione determina una serie di permessi associati a questi utenti. La decisione di autorizzazione di basa su questa lista di permessi.

Così come il servizio di autenticazione il servizio di autorizzazione prevede metodi amministrativi utili per la gestione dei permessi associati agli utenti:

```
assignPermissionToRole(OA_Permission, OA_Role) : void
assignRoleToPrincipal(OA_Role, OA_Principal) : void
createRole(OA_Role) : void
deleteRole(OA_Role) : void
getRoles(OA_Query) : Sequence<OA_Role>
unassignPermissionFromRole(OA_Permission, OA_Role) : void
unassignRoleFromPrincipal(OA_Role, OA_Principal) : void
updateRole(OA_Role) : void
```

In sostanza il sistema di autenticazione definisce per primo i ruoli (concetto mappato nell'entità *OA_Role* che descrive una tipologia di utente ad esempio "ospite" o "amministratore", ecc.) attraverso i metodi *createRole* quindi assegna dei privilegi a questi ruoli (*assignPermission*). Una volta fatto questo è possibile assegnare dei ruoli agli utenti, le entità *Principal*, che quindi riceveranno i permessi associati ai ruoli sopra definiti e tali permessi verranno controllati dal servizio di autorizzazione.

5.1.6 Feature Access Service

Il Feature Access Service (FAS) permette l'accesso in scrittura ed in lettura a istanza di feature disponibili nell' Orchestra Service Network. FAS fornisce dunque l'interfaccia che può essere ereditata anche da FAS più specifici (ad esempio il servizio di accesso ai dati di un sensore).

Il FAS offre informazioni relative a:

- I feature type che è in grado di fornire o supportare
- I formati di codifica dell'informazione supportati
- I linguaggi di interrogazione e i meccanismi supportati per l'accesso ai dati.

Il FAS permette solamente l'accesso a informazioni che sono rappresentate attraverso feature types che seguono le direttive dell' Orchestra Meta-model.

Il fatto che le informazioni siano rimodellate on-the-fly da un componente software o i dati siano immagazzinati in qualche struttura non è cruciale per il FAS: visto attraverso l'interfaccia, la rappresentazione della feature è una scatola nera che non è visibile ai client.

Il Feature Access Service permette di ricercare le features in base al loro tipo, in base a specifici valori assunti dagli attributi e in base a parametri spaziali e temporali. Le istruzioni di ricerca sono codificate utilizzando un linguaggio di ricerca che supporta tutte le funzionalità richieste (ad esempio SQL includendo istruzioni spazio-temporali).

Le istanze di feature e i feature types sono identificabili attraverso un identificatore Univoco (UID) che è univoco in almeno un OSN.

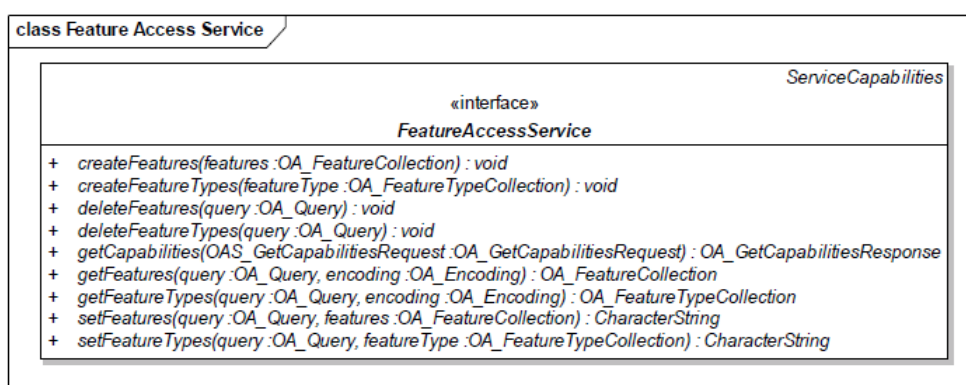


Figura 16 - Interfaccia del Servizio Feature
(RM-OA Rev2.1 as OGC Best-practice document 07-097)

L'interfaccia di accesso al servizio contiene i metodi per la creazione e l'aggiornamento di Feature come *createFeatures* e *setFeatures* mentre l'operazione *getFeatures* permette il reperimento dei contenuti da parte di altri servizi.

5.1.7 Catalogue Service

Il *Catalogue Service* supporta l'abilità di pubblicare, ricercare e recuperare informazioni descrittive per risorse come dati e servizi ma anche caratteristiche che estendono il *OMM_FeatureType* come ad esempio documenti, dizionari e modelli di equazione.

Il servizio non è legato ad un particolare schema o standard; supporta invece tutti i tipi di application schemas per meta informazioni che sono state progettati in accordo alle regole dell' Orchestra Meta Model (OMM).

Grazie all'indipendenza del servizio dalle piattaforme di implementazione, il catalogo può essere usato per salvare meta informazioni riguardanti i servizi e le risorse; quindi una istanza del servizio può essere usata come catalogo di dati, un registry di servizi o entrambi se l'istanza del servizio prevede diversi tipi di meta informazioni.

Le diverse meta informazioni presenti nel catalogo rappresentano le caratteristiche delle risorse che possono essere ricercate e presentate per una valutazione all'uso ed eventualmente una successiva rielaborazione sia software che umana.

Il servizio supporta la ricerca di una risorsa all'interno del network; se la risorsa viene trovata all'interno della comunità all'utente vengono restituite le informazioni per collegarsi alla risorsa richiesta (ad esempio viene restituito un URI).

I servizi forniti da un catalogo possono essere divisi in 4 tipologie: Ricerca, Navigazione, Pubblicazione e Collezione.

Uno dei punti di forza del catalogo risiede nell'interoperabilità sia tecnica che semantica; a tal fine il servizio catalogo è relazionato con diversi tipi di meta informazioni:

Meta informazioni incluse nel catalogo: le meta informazioni gestite dal catalogo vengono usate per scopi di ricerca di risorse e servizi. Questo non esclude la possibilità che il catalogo gestisca delle meta informazioni non direttamente inerenti alla ricerca. Le meta informazioni sono gestite nel catalogo attraverso tipi di meta informazioni

rappresentate attraverso degli standard come ad esempio OAS-MI (Orchestra meta information schema), OGC o UDDI o strutture per la gestione delle informazioni semantiche come RFD e OWL.

Meta informazioni offerte dal catalogo: L'istanza del catalogo offre meta informazione attraverso le sue *capabilities*. Le capabilities di un servizio catalogo possono assolvere a diversi scopi come ad esempio l'accesso ai dati o l'invocazione di un servizio.

Per questo il catalogo mette a disposizione 5 interfacce:

Service Capabilities Interface Type: interfaccia che permette il reperimento delle capabilities fornite al servizio catalogo.

Catalogue Search Interface Type: Attraverso l'interfaccia di ricerca il client può chiedere al catalogo la disponibilità di tipi di risorsa. Ad ogni tipo di risorsa presente in catalogo è associato alle meta informazioni sul tipo e il suo corrispondente linguaggio di interrogazione. Grazie a queste informazioni il client può interrogare il tipo di risorsa desiderato nel linguaggio appropriato.

```
search(OA_SearchRequest) : OA_SearchResponse
```

Il metodo *search* messo a disposizione da questa interfaccia innesca il processo vero e proprio di ricerca.

Catalogue Publication Interface Type: Attraverso questa interfaccia è possibile includere, aggiornare e eliminare le meta informazioni dal catalogo.

Catalogue Collection Interface Type: Questa interfaccia prevede operazioni utili per l'aggiornamento automatico del contenuto del catalogo in modo differente dalla pubblicazione.

Catalogue Management Interface Type: Questa interfaccia prevede le operazioni di configurazione da parte dell'utente di scenari in cui sono presenti più cataloghi in cascata

5.1.8 Map and Diagram Service

Il Map and Diagram Service è un servizio che dinamicamente disegna dati geografici e statistici utilizzando delle convenzioni stilistiche. Il suo compito principale consiste nel produrre mappe e diagrammi da dati geografici (vettori o raster) o da dati statistici (ad esempio censimenti, risultati di analisi statistiche) come file immagine digitale pronto per essere visualizzato sui display. Questo servizio può creare mappe e diagrammi basandosi non solo sui dati che risiedono sul server in cui è installato il servizio, ma anche basandosi su dati forniti da servizi esterni o direttamente inclusi nel messaggio di richiesta di una mappa o diagramma.

L'output principale del servizio è un documento immagine; esso può essere una mappa, ovvero la visualizzazione di informazioni geografiche e dati spaziali, un diagramma, in sostanza la visualizzazione di dati statistici, o la legenda di un livello creato.

Le mappe e i diagrammi rappresentano la visualizzazione dei dati e non sono dati essi stessi, perciò questi sono creati utilizzando dei file piatti da cui non è possibile ricostruire l'informazione iniziale che li ha generati come file il formato PNG, GIF o JPG.

Analizzando i metodi specificati dall'interfaccia del servizio i metodi principali di reperimento dei contenuti sono:

```
getMap(GetMap :OA_GetMapType) : OA_GetMapResponse  
getDiagram(GetDiagram :OA_GetDiagramType) : OA_GetDiagramResponse
```

5.2 Estensione: Gestione Eventi

Come descritto nel capitolo precedente la gestione degli eventi è stata individuata come la possibile soluzione al problema dell'aggiornamento delle risorse che compongono uno scenario: il verificarsi di una condizione associata ad una risorsa abilitata alla gestione di eventi (ad esempio la modifica di un valore, il cambiamento di

stato, il passaggio di un determinato lasso temporale) scatena l'evento che darà il via ad una serie di operazioni specificate dall'utente.

Per rendere possibile l'operazione di controllo degli stati delle risorse e la relativa notifica occorre utilizzare oggetti che mettono a disposizione interfacce per la creazione e gestione di listener seguendo il concetto architetturale dell' Observer Pattern.

Observer Pattern

L'Observer pattern è un design pattern utilizzato per tenere sotto controllo lo stato di diversi oggetti ed è quindi è utilizzato come la base architetturale di molti sistemi di gestione di eventi. Il pattern si basa su uno o più oggetti, chiamati osservatori o listener, che vengono registrati per gestire un evento che potrebbe essere generato dall'oggetto "osservato".

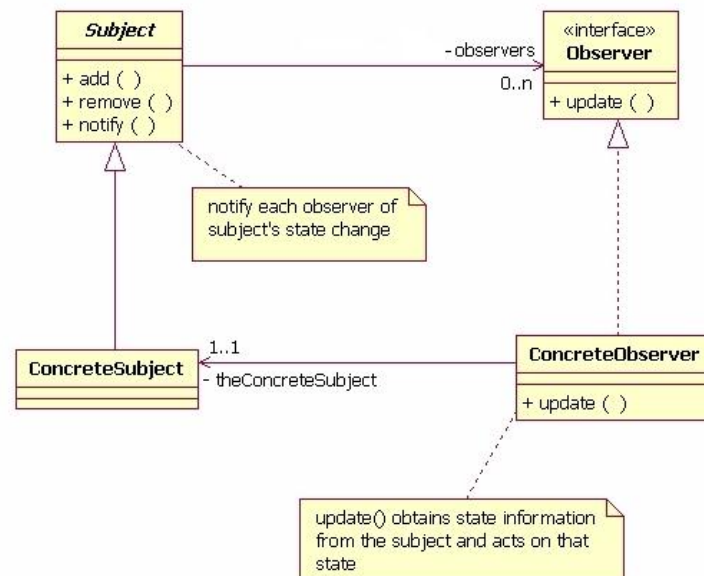


Figura 17 - Diagramma UML dell' observer pattern

Nella figura è possibile notare come l'Observer consista nella specifica di un'interfaccia astratta associabile ad un qualsiasi evento; tale interfaccia viene quindi implementata

dal ConcreteObserver il quale si occupa di effettuare delle operazioni specifiche a seconda del contesto.

Estensione dell' OA Basic Service

Analizzando la specifica di tutti i servizi descritti nell'Architettura Orchestra si è individuato l'OA Basic Service come un valido soggetto al quale poter aggiungere il concetto di evento e della sua gestione.

La scelta è ricaduta sull'OA Basic Service poiché esso è il punto di partenza per la specifica e l'implementazione di tutti gli altri servizi e quindi, inserendo il concetto di evento a questo livello di generalità, è possibile rendere event-enabled tutti i servizi facenti parte dell'OSN in quanto naturali estensioni del OA Basic Service. Utilizzando tale oggetto è stato inoltre ripreso ed ampliato il concetto di interazione asincrona tra i servizi come specificato dall'interfaccia Asynchronous Interaction. Tale interfaccia, risulta essere una solida base per la creazione di una gestione eventi in quanto prevede la creazione di chiamate asincrone che si concludono con la chiamata del metodo di risposta notify(OA_Notification) che, nonostante permetta la Callback all'oggetto chiamante, non gestisce la possibilità di specificare metodi destinatari dell'evento, permettendo la semplice comunicazione di una risposta alla chiamata asincrona precedentemente invocata.

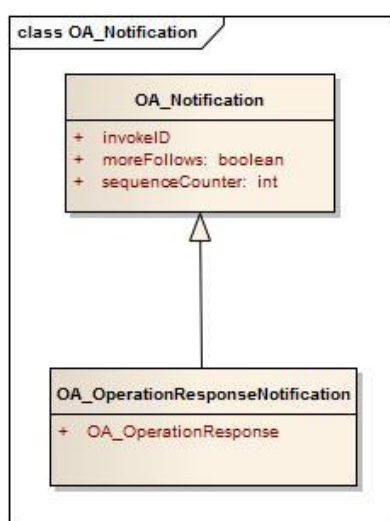


Figura 18 - Class Diagram dell' Oggetto OA_Notification

Dalla figura si nota che l'oggetto OA_Notification, usato come input del metodo che notify, è molto semplice e contiene solamente l'identificativo dell'invocazione del metodo e indica se seguiranno ulteriori invocazioni. Una possibile estensione dell'oggetto in questione è stata proposta nell'architettura Orchestra, l'OA_OperationResponseNotification, ma tale oggetto può contenere in più rispetto all'oggetto padre un eventuale valore di ritorno del metodo.

Interfaccia Service Event

L'estensione vera e propria dell'OA Basic Service si realizza attraverso l'introduzione dell'interfaccia ServiceEvent che contiene tutti i metodi di gestione dei possibili eventi come descritto dalle figure di seguito.

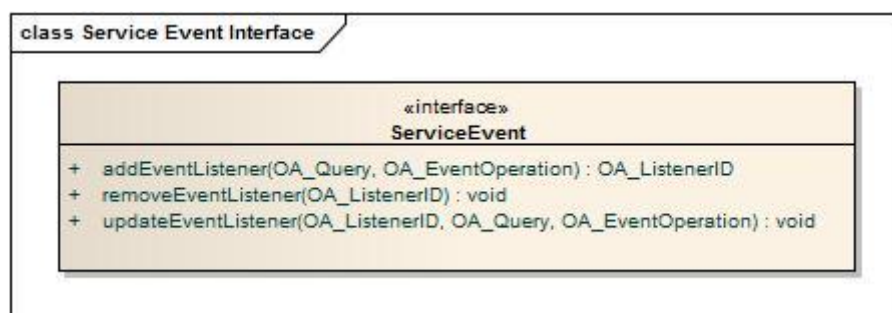


Figura 19 - Diagramma UML dell'interfaccia Event Service

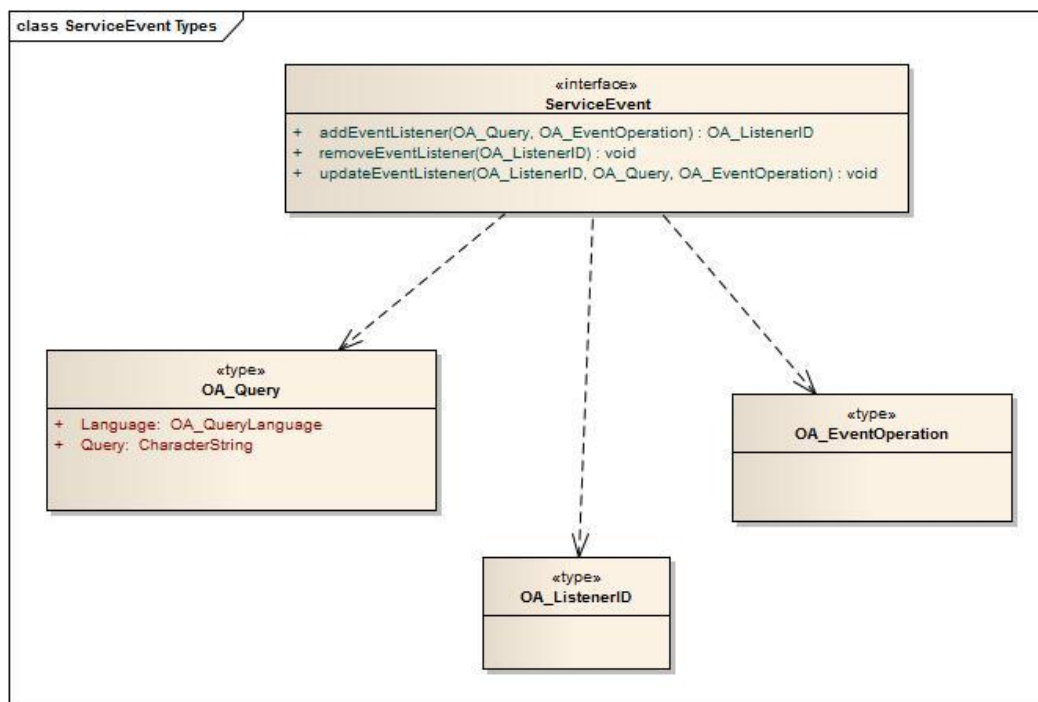


Figura 20 - OA Types dell'interfaccia Service Event

La figura rappresenta l'interfaccia di gestione degli eventi che specifica 3 metodi:

addEventListener: questo metodo permette l'attivazione del servizio di notifica dell'evento. Il metodo riceve in ingresso gli oggetti *OA_Query* e *OA_EventOperation* e restituisce l'oggetto *OA_ListenerID*.

Il parametro d'ingresso *OA_Query* indica le condizioni di attivazione dell'evento: questo può indicare diverse condizioni in base alla tipologia di servizio dal quale la funzionalità di gestione degli eventi può essere implementata. Ad esempio se il servizio è implementato da un Sensor Access Service il parametro d'ingresso potrebbe richiedere di creare l'evento ogni qual volta il valore letto da un sensore supera un determinato valore oppure di creare l'evento ogni 24 ore senza che altre situazioni si verifichino.

Il parametro d'ingresso *OA_EventOperation* invece specifica l'interfaccia del servizio da richiamare a fronte del verificarsi dell'evento: grazie a questa estensione è possibile richiamare un qualsiasi metodo esposto da un generico servizio dell'OSN. Tale

funzionalità risulta molto utile nella gestione di eventuali aggiornamenti o distribuzioni di Alert attraverso l'implementazione di specifici metodi (Capitolo 6)

Al servizio da invocare basterà implementare o estendere un generico metodo *OA_EventOperation* per rendere possibile l'invocazione dei propri servizi da parte di un oggetto event-enabled.

Il metodo ritorna l'oggetto *OA_ListenerID* che indica l'identificativo del listener creato appositamente per l'osservazione dell'oggetto desiderato. Questo identificativo può essere utilizzato per modificare o disabilitare il servizio di notifica eventi.

updateEventListener: metodo utilizzato per modificare i parametri del servizio di notifica. Al metodo oltre agli oggetti *OA_Query* e *OA_EventOperation*, che indicano le condizioni di attivazione dell'evento e le operazioni da eseguire a fronte dell'evento, viene passato l'identificativo *OA_ListenerID* che identifica il listener al quale applicare le modifiche.

removeEventListener: il metodo permette la distruzione del listener identificato dall'identificativo *OA_ListenerID* passato al metodo.

L'estensione proposta permette quindi la gestione di eventi all'interno dell'architettura Orchestra senza stravolgerla o modificarne le fondamenta grazie alla modularità propria dei Web Service.

5.3 Estensione: Concetto Scenario

Come già definito al capitolo 4, uno scenario è la composizione di alcuni servizi già disponibili all'interno dell'architettura e può contenere:

- Mappe e Dati Cartografici (Mappe, livelli di Mappe, dati rasterizzati su Mappe)
- Feature o Oggetti geo referenziati quali modelli di oggetti reali, dati associati ad un particolare punto spaziale...
- Sensori

- Documenti

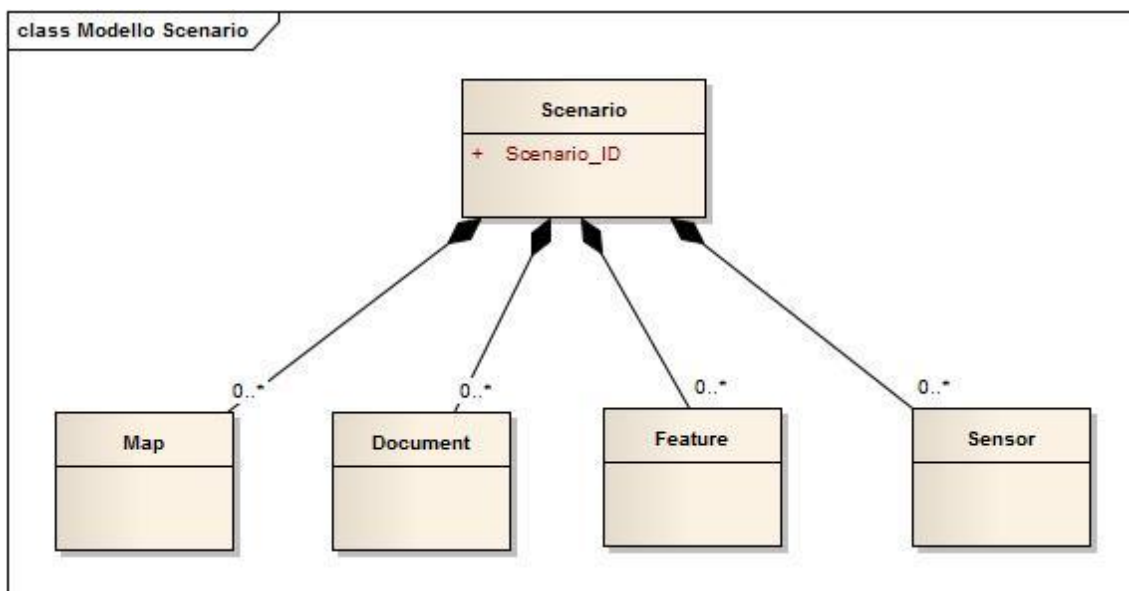


Figura 21 - Scenario come contenitore di oggetti (Servizio di Servizi)

Lo scenario può divenire esso stesso un servizio all'interno dell'architettura composto da una successione di informazioni che possono essere recuperate, modificate ed aggiornate tramite l'accesso ai diversi servizi OA Info-Structure presenti all'interno dell'architettura Orchestra.

Il servizio scenario permette la pubblicazione, la ricerca e l'interrogazione di scenari come aggregati di informazioni presenti all'interno dell'architettura Orchestra.

5.3.1 Nuovi OA_Types

Il nuovo servizio Scenario definisce alcuni nuovi OA Types

OA_ScenarioID

Il tipo OA_ScenarioID è utilizzato per definire in modo univoco un particolare scenario sull'OSN secondo quanto già definito ed applicato per i sensori od altre interfacce all'interno di Orchestra. Il contenuto di tale identificativo è dipendente dall'implementazione.

OA_ScenarioDescriptor

Il tipo OA_ScenarioDescriptor è una rappresentazione del tipo di scenario, attraverso un nome, una descrizione e altri dati utili al suo corretto utilizzo.

OA_ScenarioData

Il tipo OA_ScenarioData definisce il formato con cui il servizio scenario restituisce i dati. I dati descritti dall'oggetto OA_ScenarioData sono la "fotografia" dello scenario così come presente all'interno dell'architettura.

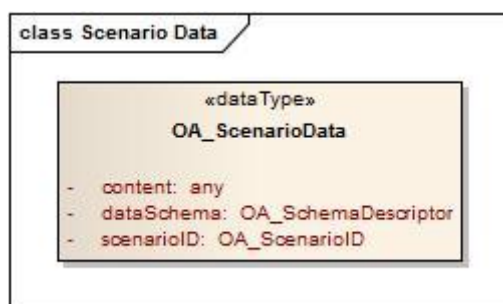


Figura 22 - OA_ScenarioData - dataType

Ipotesi implementativa di OA_ScenarioData

Nel frammento di codice seguente si ipotizza una definizione non completa e non formale attraverso il meta-linguaggio XML al fine di esemplificare quanto descritto.

```
<scenarioData>
  <ScenarioID><!-- OA_ScenarioID --><ScenarioID>
  <dataSchema><!-- OA_SchemaDescriptor -->< dataSchema >

  <scenarioObjects>
    <scenarioObject type="layer" eventDriven="true|false">
      <position> <!-- geo position --> </position>
      <lastUpdate> <!-- geo position --> </lastUpdate>
      <!-- JPEG/PNG content Data -->
    </scenarioObject>

    <scenarioObject type="feature" eventDriven="true|false">
      <position> <!-- geo position --> </position>
      <lastUpdate> <!-- geo position --> </lastUpdate>
      <!-- GML content Data -->
    </scenarioObject>
  </scenarioObjects>
</scenarioData>
```

```

</scenarioObject>

<scenarioObject type="sensor" eventDriven="true|false">
  <position> <!-- geo position --> </position>
  <lastUpdate> <!-- geo position --> </lastUpdate>
  <!-- GML content Data -->
</scenarioObject>

<scenarioObject type="document" eventDriven="true|false">
  <position> <!-- geo position --> </position>
  <lastUpdate> <!-- geo position --> </lastUpdate>
  <!-- PDF/DOC content Data -->
</scenarioObject>
</scenarioObjects>
</scenarioData>

```

OA_ScenarioConfiguration

Descrive la configurazione ed il processo che ha portato alla realizzazione dello scenario in termini di sorgenti delle informazioni, definizione dei listener per gli eventi associati alle informazioni stesse, query e altre informazioni utili alla sua creazione.

Può essere considerato il progetto dello scenario, in quanto permette a qualunque altro nodo dell'architettura di ricreare lo scenario effettuando in cascata tutte le operazioni descritte.

La differenza con l'oggetto OA_ScenarioData sta nella diversa finalità e concezione degli oggetti. OA_ScenarioData è una fotografia dello scenario in un determinato momento, OA_ScenarioConfiguration è la descrizione delle operazioni che hanno reso possibile la creazione dello scenario e la conseguente fotografia.

Ipotesi implementativa di OA_ScenarioConfiguration

Nel frammento di codice seguente si ipotizza una definizione non completa e non formale attraverso il meta-linguaggio XML al fine di esemplificare quanto descritto.

```

<scenarioConfiguration>

```

```

<scenarioDefinition>
  <scenarioObject type="" id="" eventDriven="true|false">
    <informationAddress><informationAddress>
      <informationSchema></informationSchema>
    </scenarioObject>
  <scenarioObject type="" id="" eventDriven="true|false">
    <informationAddress><informationAddress>
      <informationSchema></informationSchema>
    </scenarioObject>
  <scenarioObject type="" id="" eventDriven="true|false">
    <informationAddress><informationAddress>
      <informationSchema></informationSchema>
    </scenarioObject>
</scenarioDefinition>
<attachment></attachment>
<creator></creator>
</scenarioConfiguration>

```

OA_ScenarioObject

Il tipo OA_ScenarioObject identifica in modo generale un qualunque oggetto presente all'interno dello scenario, sia esso una mappa, un layer, un oggetto, un documento etc...

```

<informationAddress><informationAddress>
  <informationSchema></informationSchema>
</position> <!-- geo position --> </position>
  <lastUpdate> <!-- geo position --> </lastUpdate>

type="" id="" eventDriven="true|false">

```

5.3.2 Eccezioni

Si riportano di seguito alcune Eccezioni che il nuovo servizio può sollevare già definite all'interno del progetto Orchestra.

- **OA_InvalidParameterValue:** Operation request contains an invalid parameter value. Return the name of the parameter with invalid value.
- **OA_MissingParameterValue:** Operation request does not include a parameter value. Return the name of the missing parameter.
- **OA_NoApplicableCode:** No other basic or service-specific exception type applies.
- **OA_InternalError:** A problem occurred in the runtime environment (e.g. out of memory)
- **OA_IllegalQuery:** Operation request does not contain a valid query

5.3.3 Interfaccia ServiceCapabilities

Interfaccia obbligatoria del servizio che fornisce la descrizione delle funzionalità supportate.

getCapabilities

Restituisce al richiedente le informazioni riguardo le funzionalità messe a disposizione dal servizio Scenario. Alcuni esempi di queste informazioni possono riguardare il linguaggio di query dei dati, meta information types utilizzati nel servizio catalogo e la struttura dei dati. Di seguito si definiscono due nuovi parametri legati al concetto di scenario definito:

eventSupported:Boolean: identifica il supporto all'architettura ad eventi dell'interfaccia

historyEnabled:Boolean identifica il supporto alla storicizzazione degli scenari

5.3.4 Interfaccia ScenarioAdministration

Interfaccia per l'amministrazione dello scenario: creazione, modifica, eliminazione

addScenarioDescription

Aggiunge uno scenario al servizio attraverso la sua descrizione. L'aggiunta della descrizione dello scenario comporta la pubblicazione dello stesso, fatto salvo la gestione del livello autorizzativo. Il valore di ritorno identifica univocamente lo scenario all'interno dell'architettura.

```
OA_ScenarioID addScenarioDescription ( OA_ScenarioDescriptor ) throws  
OA_InvalidParameterValue, OA_MissingParameterValue, OA_NoApplicableCode,  
OA_InternalError
```

removeScenarioDescription

Elimina lo scenario dal servizio e di conseguenza la possibilità di utilizzo e la sua pubblicazione

```
void removeScenarioDescription ( OA_ScenarioID ) throws  
OA_InvalidParameterValue, OA_MissingParameterValue, OA_NoApplicableCode,  
OA_InternalError, OA_InvalidScenarioID
```

updateScenarioDescription

Aggiorna il descrittore dello scenario.

```
void updateScenarioDescription ( OA_ScenarioID, OA_ScenarioDescriptor )  
throws OA_InvalidParameterValue, OA_MissingParameterValue,  
OA_NoApplicableCode, OA_InternalError, OA_InvalidScenarioID
```

setScenarioData

Metodo finalizzato all'update dei dati dello Scenario. Il metodo esposto permette l'aggiornamento di tutti i componenti presenti all'interno dello scenario (Mappe, Feature, Sensori...). Il metodo può essere utilizzato sia dall'editore dello scenario, sia

dai servizi che implementano le funzionalità di gestione dell'evento presso i quali lo scenario si è registrato. Al fine di permettere l'utilizzo del metodo *setScenarioData* nella gestione di eventi si ipotizza che lo stesso implementi *OA_EventOperation* precedentemente descritto

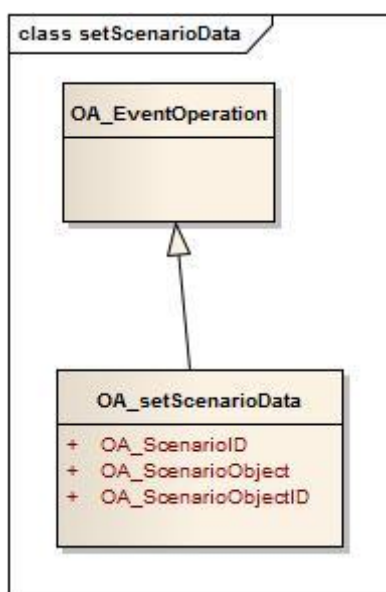


Figura 23 - *setScenarioData* come *OA_EventOperation*

```
void setScenarioData ( OA_ScenarioID, OA_ScenarioObjectID,  
OA_ScenarioObject ) throws OA_InvalidParameterValue,  
OA_MissingParameterValue, OA_NoApplicableCode, OA_InternalError,  
OA_InvalidScenarioID, OA_InvalidScenarioObjectID
```

Al fine di utilizzare con successo la gestione degli eventi, qualora l'utente desiderasse l'aggiornamento dei dati da parte di un particolare oggetto inserito in uno scenario, dovrà registrarsi presso lo stesso comunicando:

- Query per la scelta degli eventi per i quali restare in ascolto
- metodo *setScenarioData* al fine di permettere l'aggiornamento dei dati da parte dell'oggetto richiesto.

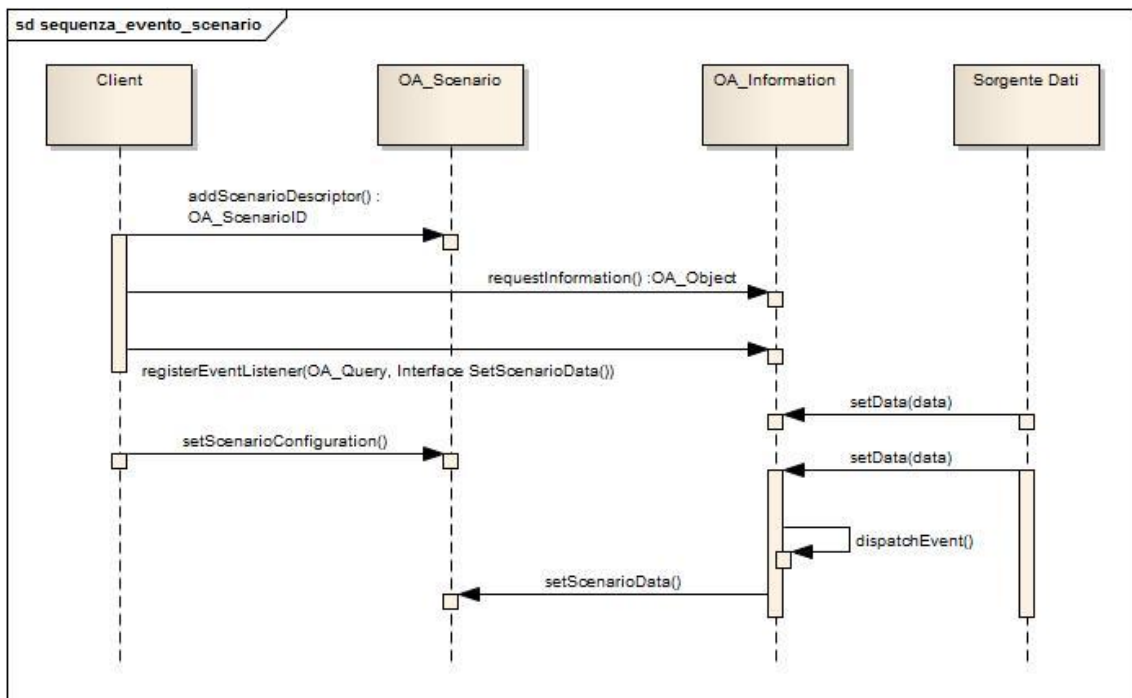


Figura 24 - Registrazione e notifica di Evento tra Scenario e Informazione Generica

Nel Sequence Diagram precedente è evidenziato il processo di gestione degli eventi tra oggetti e scenario:

- L'utente richiede un particolare OA_Object
- Decide di registrarsi presso l'oggetto per la notifica di eventi passando la query per la definizione di quali eventi gestire e il metodo che l'oggetto dovrà richiamare al verificarsi dell'evento.
- L'utente pubblica lo scenario
- Al verificarsi dell'evento, l'oggetto chiamerà il metodo setScenarioData, per l'aggiornamento del dato.

5.3.5 Interfaccia ScenarioConfiguration

setScenarioConfiguration

Definisce la configurazione di uno scenario definendo gli oggetti e i layer che lo compongono. A differenza di OA_ScenarioData, OA_ScenarioConfiguration non

contiene il dato reale ma la descrizione sulla composizione ed il reperimento delle informazioni a livello di interfacce dell'architettura usate e loro interazione.

```
void setScenarioConfiguration ( OA_SensorID, OA_ScenarioConfiguration )  
throws  
OA_InvalidParameterValue, OA_MissingParameterValue, OA_NoApplicableCode,  
OA_InternalError, OA_InvalidScenarioID
```

getScenarioConfiguration

Restituisce la configurazione dello scenario con i riferimenti delle singole interfacce usate per la creazione dello stesso

```
OA_ScenarioConfiguration getScenarioConfiguration ( OA_SensorID ) throws  
OA_InvalidParameterValue, OA_MissingParameterValue, OA_NoApplicableCode,  
OA_InternalError, OA_InvalidScenarioID
```

5.3.6 Interfaccia ScenarioData

L'interfaccia Scenario Data permette l'interrogazione e il recupero delle informazioni di scenario esposte attraverso il servizio. Le informazioni che descrivono un servizio sono composte sia dai dati elaborati (mappe, oggetti geo referenziati, documenti) sia dalle informazioni per il recupero dei dati stessi.

A tal proposito l'architettura prevede due diverse funzionalità per l'accesso allo scenario che restituiscono però la stessa tipologia di output (OA_ScenarioData). La scelta di separare le due funzionalità ricade nella diversa finalità del dato in uscita e pertanto dalla gestione di eventuali livelli autorizzativi.

getScenarioData

Restituisce i dati attuali presenti sullo scenario in termini di mappe, sensori, funzionalità.

```
Sequence <OA_ScenarioData> getScenarioData ( OA_Query ) throws  
OA_InvalidParameterValue, OA_MissingParameterValue, OA_NoApplicableCode,  
OA_InternalError, OA_IllegalQuery, OA_InvalidScenarioID
```

getScenarioHistory

Restituisce uno o più scenari sulla base di una query temporale. I dati storici dello scenario possono essere richiesti solo se lo scenario implementa la capability `historyEnabled`.

La funzionalità permette di ottenere

- scenario singolo registrato in un singolo istante temporale
- più scenari in sequenza registrati in un determinato intervallo temporale
- uno o più scenari registrati e selezionati a partire da alcune caratteristiche dello scenario di partenza (es. particolari dati sensoristici o feature)
- più scenari in sequenza registrati a partire da uno scenario di partenza per un determinato lasso temporale.

```
Sequence <OA_ScenarioData> getScenarioHistory ( OA_Query ) throws  
OA_InvalidParameterValue, OA_MissingParameterValue, OA_NoApplicableCode,  
OA_InternalError, OA_IllegalQuery, OA_InvalidScenarioID
```

5.3.6 Interfaccia ScenarioEvent

L'interfaccia `ScenarioEvent` è l'estensione per il servizio Scenario dell'interfaccia di gestione Eventi dell'`OA_BasicService`

addScenarioEventListener

Permette l'aggiunta di un Listener a cui notificare il verificarsi di un evento generato dallo scenario.

- `OA_Query`: Query che identifica l'evento per il quale il Listener si registra
- `OA_EventOperation()`: Riferimenti dell'operazione da chiamare al verificarsi dell'evento.
- `OA_ListenerID`: identificativo del Listener, da utilizzare per l'identificazione del Listener a livello di Scenario

```
OA_ListenerID addScenarioEventListener ( OA_Query, OA_EventOperation() )  
throws OA_InvalidParameterValue, OA_MissingParameterValue,  
OA_NoApplicableCode, OA_InternalError, OA_IllegalQuery
```

updateScenarioEventListener

Permette la modifica di un Listener già creato

```
void updateScenarioEventListener (OA_ListenerID, OA_Query,  
OA_EventOperation() )  
throws OA_InvalidParameterValue, OA_MissingParameterValue,  
OA_NoApplicableCode, OA_InternalError, OA_IllegalQuery
```

removeScenarioEventListener

Permette l'eliminazione di un Listener

```
void removeScenarioEventListener (OA_ListenerID) throws  
OA_InvalidParameterValue, OA_MissingParameterValue, OA_NoApplicableCode,  
OA_InternalError
```

Capitolo 6

Scenari di Validazione

Scopo del presente capitolo è presentare le peculiarità dell'architettura utilizzando due scenari di validazione: il Caso Lambro ed il Caso Premana. All'interno del capitolo si descrive utilizzando esempi pratici l'utilizzo dell'architettura, con particolare attenzione alle estensioni proposte in questo documento.

6.1 La Validazione dell'architettura

Il processo di validazione dell'architettura estesa del nuovo concetto di Scenario si è concentrato su due casi di rischio differenti. Lo scenario di rischio legato all'esondazione del Fiume Lambro a diversi livelli di dettaglio e il movimento franoso di materiale roccioso nel comune di Premana.

Il caso di esondazione del Fiume Lambro identifica uno scenario di rischio legato ad una data ciclicità dell'evento, in una zona fortemente urbanizzata sulla quale si trovano diversi insediamenti o realizzazioni umane che determinano criticità (centri commerciali, viabilità, scuole etc..). Con questo caso si intende presentare e validare la funzionalità di condivisione dei dati, di creazione di uno scenario, di aggiornamento dei dati di visualizzazione di uno scenario già creato e di ricerche all'interno di dati storici.

Il caso di Premana identifica invece uno scenario di rischio contenente alcune simulazioni realizzate sulla base di studi geologici. L'area è abbastanza ristretta e presenta alcuni insediamenti urbani (centrale elettrica, polo produttivo etc..). Con

questo caso si intende presentare e validare la funzionalità di recupero di una configurazione di scenario e di notifica di eventi legati ad uno scenario precedentemente realizzato.

6.2 Il caso Lambro

Il fiume Lambro nasce poco a Nord del monte del Ghisallo e proseguendo attraverso la Valassina diventa, dopo aver attraversato la Brianza, il Milanese ed il Lodigiano, un affluente di sinistra del Po.

Fiume tra i più inquinati d'Italia, ha visto la sua situazione migliorare dal 2005 grazie all'entrata in funzione del sistema integrato di depurazione dei reflui fognari di Milano.

Fiume di non grandissima portata, è divenuto famoso insieme al Seveso per le sue ripetute esondazioni nel periodo autunnale e piovoso, tra le quali una delle più imponenti avvenuta nel novembre 2002. L'alta urbanizzazione della zona non permette infatti un corretto flusso delle acque che vengono convogliate con troppa facilità nel letto del fiume il quale in caso di piogge persistenti è incapace di smaltirle, anche a causa di una sua natura torrentizia, conosciuta già nel passato e che ha portato alla creazione di alcuni accorgimenti nel corso degli anni (prima fra tutta la gestione del Lago di Pusiano tramite il cavo Diotti). (Provincia di Milano - Settore Protezione Civile)

6.2.1 Creazione dello Scenario: caso pratico

Nel paragrafo si intende descrivere le diverse attività che portano alla creazione di uno scenario di rischio nel contesto relativo al fiume Lambro da parte di un utente esperto del dominio applicativo.

L'utente che realizza lo scenario di rischio opera su un sistema, presumibilmente un GIS, che sarà stato opportunamente connesso ad un Orchestra Service Network (OSN) e che quindi implementerà tutte le interfacce relative all'accesso e alla fruizione dei

servizi presenti nella rete; si presuppone inoltre che l'architettura informatica utilizzata sia quella descritta nel capitolo precedente, ovvero un'architettura che si basa sul modello Orchestra ma lo estende grazie all'introduzione dei concetti di Scenario di Rischio e di gestione degli eventi.

Il sistema utilizzato essendo connesso all'OSN conoscerà alcuni punti di accesso ai servizi base quali il servizio di autenticazione e di catalogo dal quale recuperare le risorse desiderate e il servizio di pubblicazione degli scenari.

Nel caso l'utente volesse accedere a risorse ancora sconosciute dal sistema verrà utilizzato il servizio catalogo attraverso il quale l'applicativo cercherà di recuperare i servizi di accesso a tali risorse in modo trasparente ma, qualora vi fossero più alternative tra le quali scegliere, chiederà all'utente di decidere a quale servizio appoggiarsi.

Il processo di creazione e di pubblicazione dello scenario si comporrà di diverse fasi che andranno dalla ricerca delle risorse situate nella zona studiata, all'abilitazione alla notifica di eventi associati alle risorse trovate alla pubblicazione dello scenario stesso nel network rendendolo così accessibile dall'esterno.

Login e inizio creazione dello scenario

L'utente accede al sistema GIS e accede alla funzione di composizione dello scenario. L'utente dovrà inserire in questa fase la descrizione dello scenario che andrà a comporre e gli obiettivi che si prefigge. Una volta ottenute queste informazioni il sistema effettuerà il login al Service Network e successivamente chiamerà per la prima volta il servizio di registrazione dello scenario.

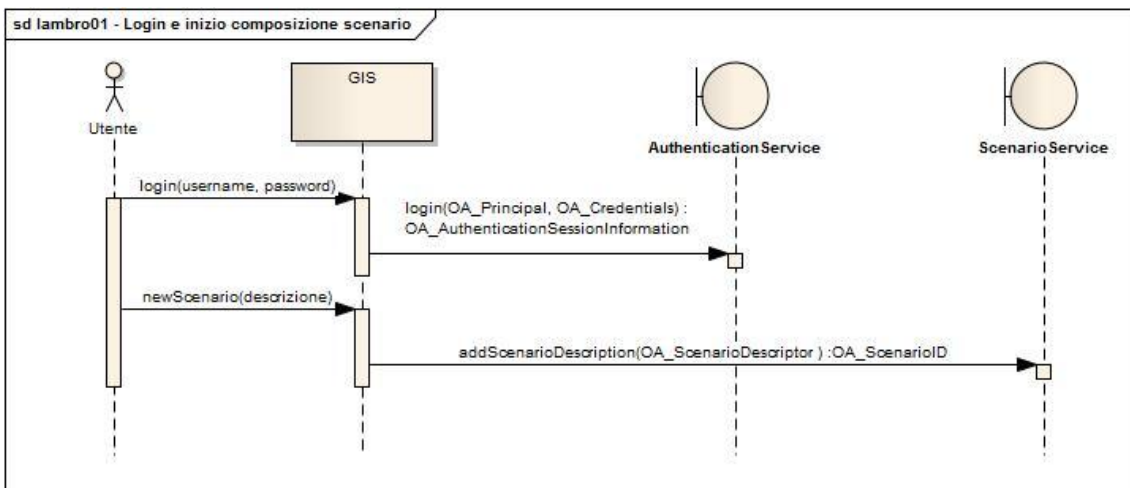


Figura 25- Login e inizio composizione scenario

Dal diagramma si nota come a seguito della chiamata della funzionalità *addScenarioDescription* il sistema come risultato della chiamata riceve l'oggetto *OA_Scenario_ID* che identifica in modo univoco in tutto il network lo scenario in questione. Tale ID sarà usato successivamente dal sistema per effettuare le operazioni di pubblicazione e di aggiornamento dei dati dello scenario.

Ricerca della mappa della zona interessata

L'utente dopo aver correttamente definito le informazioni dello scenario, inizia a richiedere al sistema tutti i dati necessari alla composizione della situazione presa in considerazione; egli interagirà sempre e solo con il sistema GIS e sarà quest'ultimo ad occuparsi del reperimento delle informazioni desiderate attraverso il network in modo trasparente.

Come prima operazione sono richieste le informazioni geografiche della zona interessata dal processo di creazione dello scenario ovvero l'area milanese attraversata dal fiume Lambro; tali informazioni sono restituite all'utente sotto forma di mappa che l'utente può visualizzare direttamente sul proprio monitor.

Si ipotizza che il sistema non disponga di tutte le informazioni richieste né conosca un servizio di accesso alle mappe; esso utilizzerà in prima battuta il Catalogue Service per reperire l'end-point del Map and Diagram Access Service che potrà soddisfare la

richiesta dei dati in questione, quindi chiamerà direttamente il servizio trovato per recuperare la mappa.

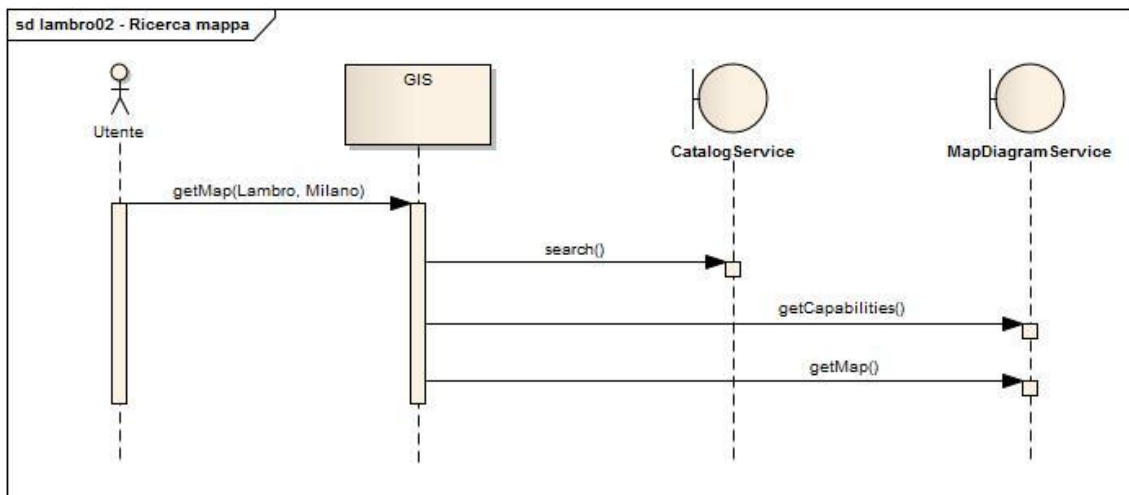


Figura 26- Ricerca Mappa

Si nota dal grafico come il sistema prima ricerchi il servizio desiderato per mezzo del servizio catalogo: L'oggetto *OA_SearchRequest* conterrà tutte le informazioni per la descrizione formale del servizio desiderato. Il sistema elaborerà l'oggetto *OA_SearchResponse* dal quale reperire le informazioni sulle modalità e l'indirizzo di interazione con il Map Access Service. Una volta processata la configurazione del servizio il sistema prima richiede le "Capabilities" al servizio target per conoscere in dettaglio tutte le funzioni supportate dal servizio. Se le funzionalità descritte soddisfano le richieste applicative e dell'utente, il sistema recupera i dati spaziali attraverso la chiamata al metodo *GetMap*.

Ricerca layer aggiuntivi: altimetria e composizione del terreno

Dopo la visualizzazione della mappa della zona interessata l'utente ritiene necessaria anche la consultazione di informazioni geografiche più dettagliate dell'area, come informazioni relative all'altimetria del terreno per valutare la presenza di avvallamenti in cui le acque potrebbero confluire e dati più precisi sulla composizione del terreno. Il sistema risponde alle richieste proponendo alcuni layer che possono essere sovrapposti alla mappa della zona.

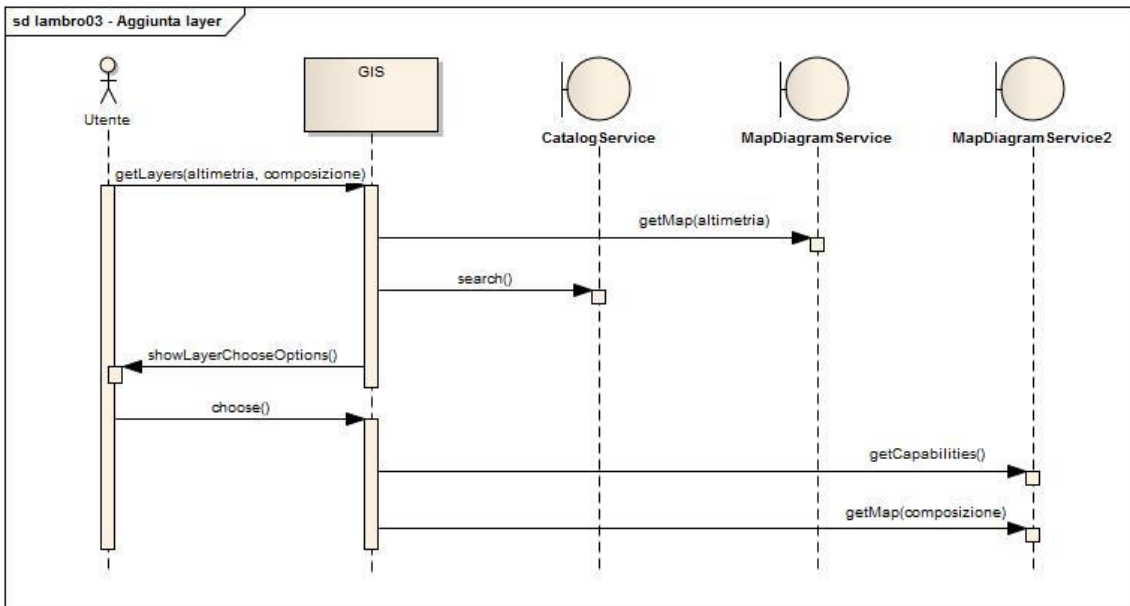


Figura 27- Aggiunta layer tematici alla mappa

Dal diagramma vediamo che il sistema richiede le informazioni relative all'altimetria al Map Access Service precedentemente interrogato per il reperimento della mappa. A questo punto il sistema è a conoscenza che questo servizio non può fornirgli le informazioni sulla composizione del terreno: esso deve quindi ricercare nel catalogo un servizio che può soddisfare la richiesta e quindi far seguire la fase di analisi delle capabilities e di richiesta della mappa.

E' possibile notare come il sistema richieda all'utente di scegliere a quale servizio di reperimento interfacciarsi sulla base delle meta-informazioni ricevute.

Utilizzo del servizio di simulazione: valutazione delle zone a rischio

L'utente decide di invocare un servizio di simulazione esterno, al fine ottenere un file grafico (un ulteriore layer) relativo alla mappa della zona, risultato della valutazione delle zone a rischio esondazione nel caso si verificasse un evento idrogeologico.

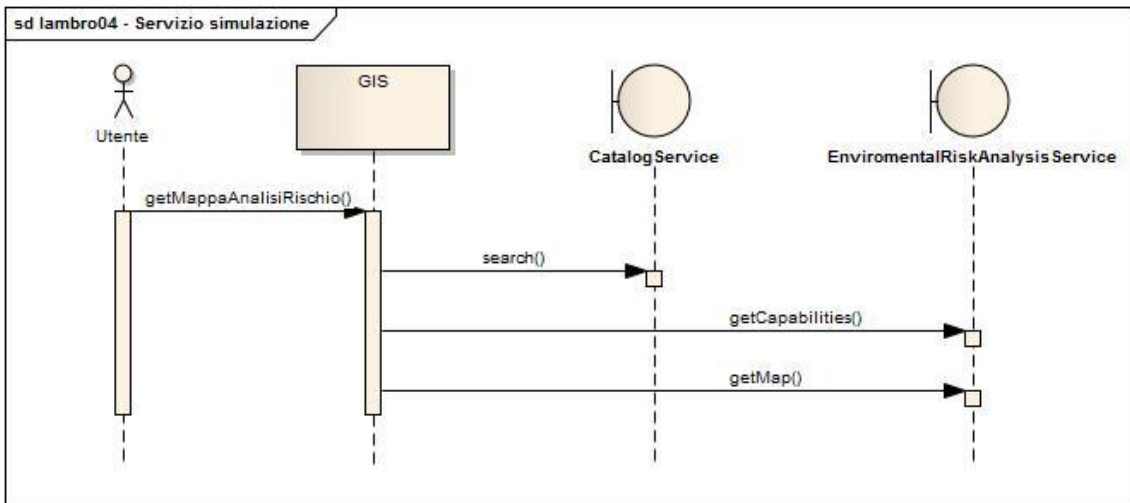


Figura 28- Servizio Simulazione e Analisi del rischio

Come nel caso precedente si nota come il sistema interagisca con il servizio catalogo per reperire le informazioni e l'ubicazione del nuovo servizio da interrogare; dopo aver richiesto ed ottenuto le informazioni desiderate il GIS mostrerà all'utente il file layer risultato dell'elaborazione.

Visione unificata della mappa

L'esperto di dominio attraverso le funzionalità del GIS sovrappone alla mappa della zona che sta valutando i veri layer ottenuti dall'interazione con gli altri servizi: in questo modo egli potrà analizzare diversi aspetti del territorio considerato al fine di valutare al meglio tutti gli possibili effetti e le zone più colpite di un eventuale evento idrogeologico causato ad esempio da una eccessiva caduta di pioggia con conseguente pericoloso innalzamento del livello del fiume Lambro.

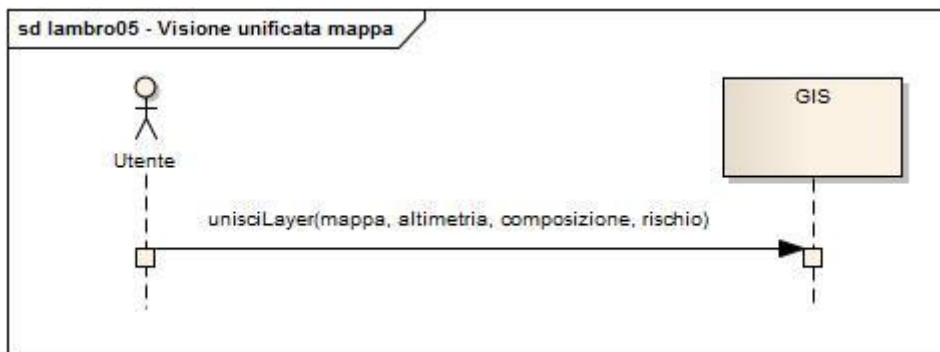


Figura 29 - Visione unificata della mappa

In questo caso l'utente interagisce solamente con il sistema sul quale sta operando in quanto le mappe utilizzate sono già state chieste nelle fasi precedenti.

Ricerca dei sensori idrometrici

L'esperto di dominio continua nella composizione dello scenario di rischio aggiungendo i dati delle rilevazioni effettuate da sensori idrometrici situati nell'area di interesse: egli richiede al sistema la ricerca dei sensori presenti nell'area coperta dalla mappa visualizzata che mettano a disposizione i dati di livello del fiume. Il sistema riporterà direttamente sulla mappa l'ubicazione dei singoli sensori ed il valore letto in fase di primo accesso al servizio.

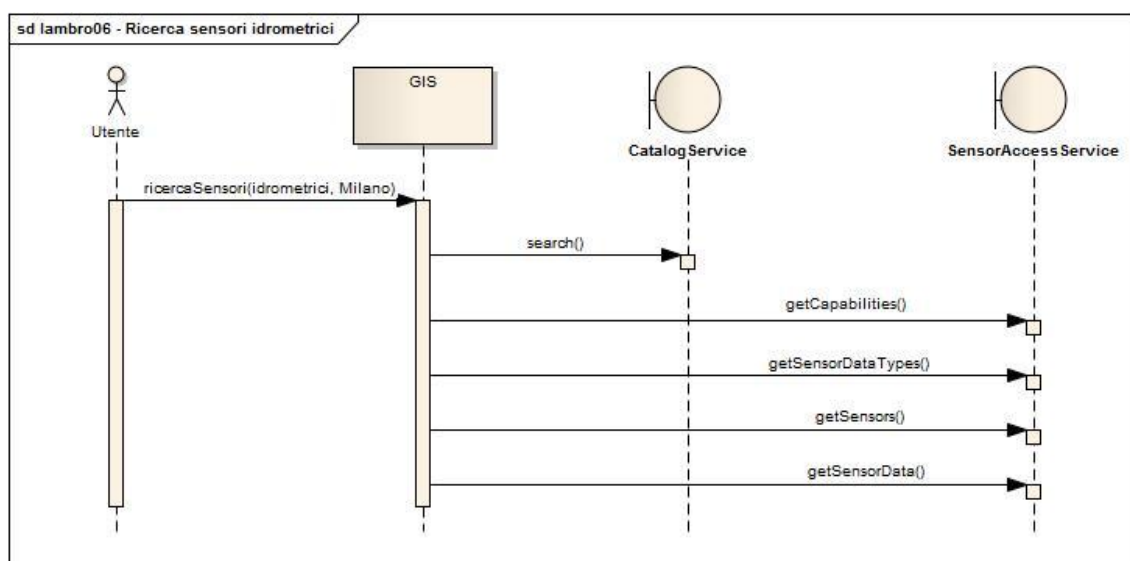


Figura 30 - Ricerca sensori idrometrici

Dal flusso si può notare come il sistema prima richieda al catalogo tutti quei servizi di accesso ai sensori che rispettano le condizioni di territorialità imposte dalla ricerca dell'utente quindi chiede al Sensor Access Service la lista dei sensori disponibili e i tipi di dato che verranno scambiati; prima di presentare il risultato dell'interrogazione del servizio all'utente il sistema richiederà il dato aggiornato ai sensori individuati attraverso il metodo *getSensorData*.

Recuperati tutti i valori di interesse l'applicativo mostrerà diversi segni grafici sulla mappa della zona che indicheranno la presenza del sensore idrometrico.

Ricerca di altri sensori a monte dell'area di interesse

L'operatore reputa necessaria la conoscenza del valore del livello del fiume di una zona più a monte rispetto a quella analizzata. Il sistema come nel caso precedente andrà a recuperare i sensori indicati e li posizionerà in una zona esterna alla mappa in quanto i sensori recuperati sono esterni alla zona di analisi.

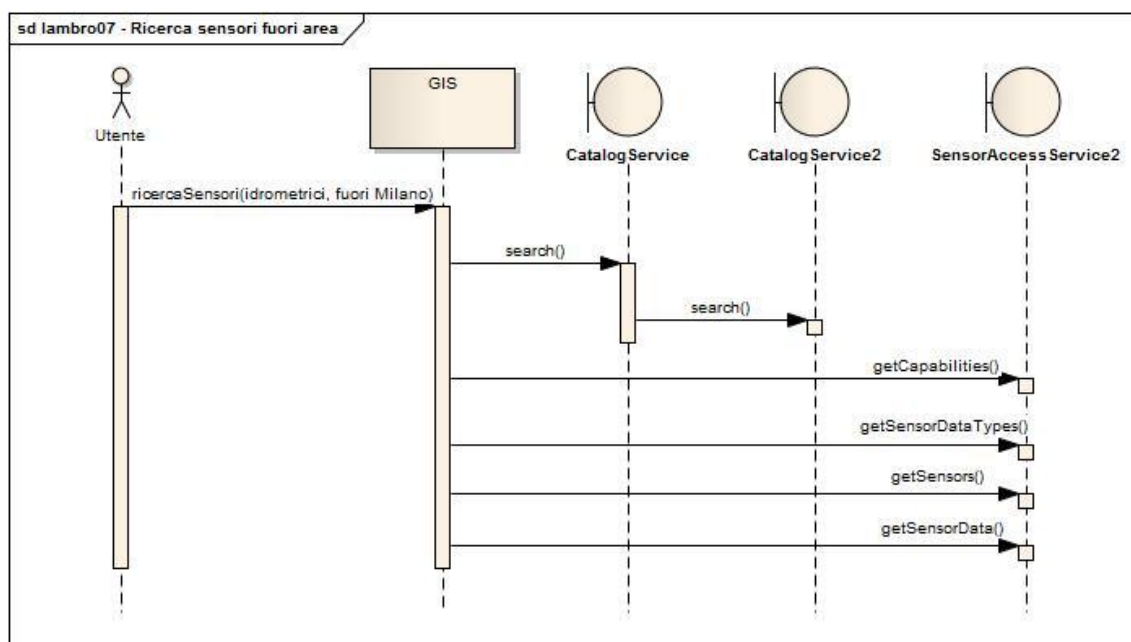


Figura 31 - Ricerca sensori idrometrici

Da questo schema si può notare la ricerca ricorsiva del servizio catalogo: si suppone che i sensori che l'utente sta cercando siano situati in una provincia diversa da quella

di Milano e che il servizio catalogo a cui il sistema si interfaccia conosca i riferimenti dei servizi di accesso ai soli sensori situati nella provincia di competenza; il catalogo quindi, non potendo soddisfare la richiesta, inoltra la query di ricerca in modo ricorsivo ad un servizio catalogo a cui esso è collegato. Questo Servizio Catalogo 2 effettuerà la stessa ricerca nel suo registry interno e restituirà al Catalogo 1 l'elenco dei servizi richiesti, altrimenti inoltrerà la richiesta a sua volta ad un altro servizio catalogo.

Una volta ricevuto l'elenco dei sensori il sistema li mostra sulla mappa come nel caso precedente.

Attivazione del servizio di notifica

L'utente procedendo con la composizione e l'analisi dello scenario di rischio decide che la conoscenza dei dati aggiornati dei livelli idrometrici e un eventuale superamento di un valore di soglia sono critici per una corretta gestione del rischio. Egli vuole essere avvisato se il livello idrometrico registrato dai sensori supera una certo valore di soglia che viene valutato come critico; il sistema attiverà il meccanismo di notifica eventi e darà conferma all'utente dell'avvenuta registrazione.

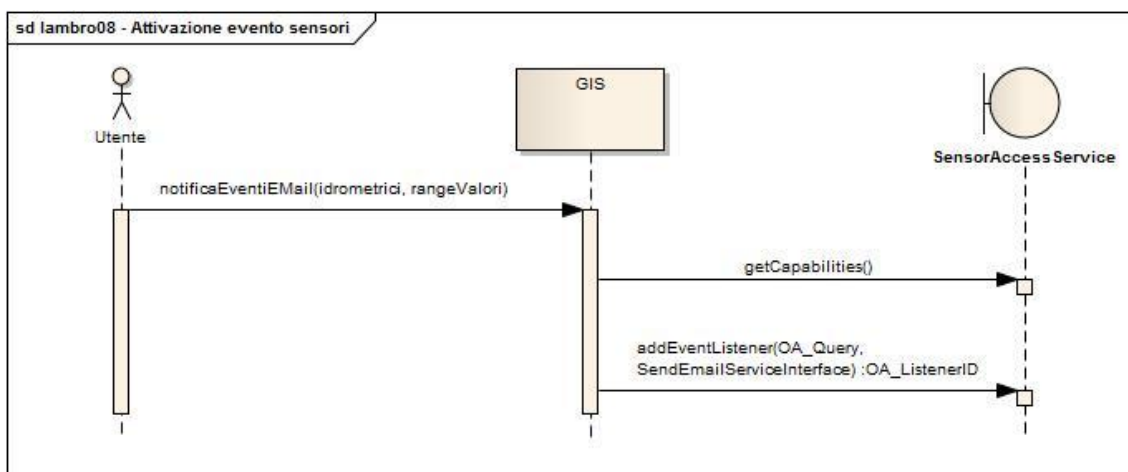


Figura 32 - Attivazione evento

Il sistema per conoscere se è possibile attivare il servizio di notifica eventi chiama la funzione *getCapabilities*: se il valore di *eventEnabled* è vero allora può avanzare la richiesta di notifica dell'evento.

L'attivazione del servizio di notifica avviene per mezzo della funzione *addEventListener*: il servizio invocato riceverà come parametri in ingresso un oggetto di tipo *OA_Query* che istruirà il servizio sulle condizioni di notifica dell'evento, mentre *OA_EventOperation* indica l'operazione da richiamare nel caso tale evento si verificasse. L'oggetto *OA_EventOperation* contiene l'end point e il nome del servizio da richiamare e i vari parametri da inviare al servizio: nel caso di analisi si suppone che l'utente voglia essere informato via e-mail dell'innalzamento del livello del fiume; in questo caso l'oggetto di richiesta conterrà i riferimenti all'interfaccia di accesso all'ipotetico servizio *SendEmailService*.

Una volta attivato il servizio il sistema ottiene un identificativo del listener e grazie al quale può registrare l'avvenuta abilitazione del servizio di notifica.

Si analizza ora il caso in cui l'evento a cui l'utente si è registrato si verifica: il servizio stesso si occuperà di richiamare la funzione specificata durante la fase di attivazione del servizio di notifica.

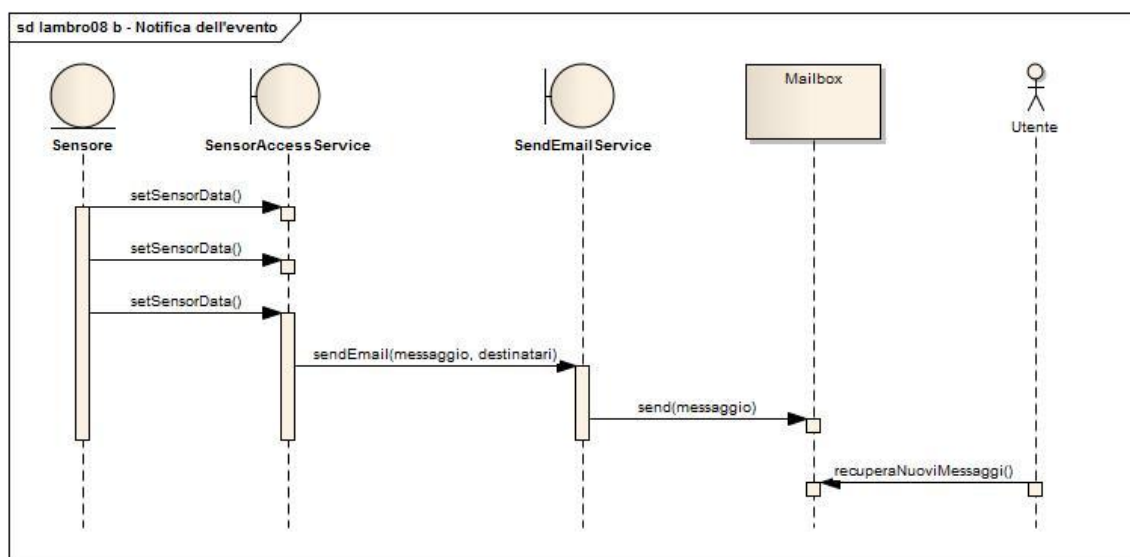


Figura 33 - Notifica dell'evento

Si vede dal diagramma come sia il servizio di accesso ai sensori ad attivarsi per notificare la condizione richiesta dall'utente; si osservi che il servizio attiva la funzionalità di notifica solo dopo la terza lettura del valore del sensore, ricevuto per

mezzo della chiamata del metodo *setSensorData*; si deduce che il valore indicato nel terzo aggiornamento rientri nel range di valori controllo dichiarati dall'utente: il *SensorAccessService* chiama l'interfaccia del servizio che ha ricevuto in ingresso durante la fase di attivazione della funzionalità; in questo caso il servizio chiama *SendEmailService* che si occuperà di inviare l'email agli indirizzi richiesti.

Ricerca di sensori meteorologici

Un fattore scatenante di una situazione di rischio idrogeologico può essere l'intensità delle precipitazioni atmosferiche; l'utente cerca nel sistema Orchestra Enabled la presenza di eventuali sensori meteorologici, in particolare la presenza di pluviometri che indicano la quantità di acqua caduta per unità di tempo.

Come nel caso dei sensori idrometrici il sistema ricercherà tutti i sensori disponibili e li mostrerà all'utente sulla mappa in base alla loro reale ubicazione e ne indicherà il dato letto in fase di chiamata del servizio.

Non è necessario introdurre nuovi diagrammi in quanto le funzionalità di discovery e binding dei sensori interessati sono già stati descritti nei diagrammi precedenti

Attivazione del servizio di notifica dei sensori

L'utente vuole essere avvisato se il volume di precipitazione per unità di tempo supera un certo valore che egli ritiene critico, come ad esempio se il livello della pioggia supera 5 mm di precipitazioni all'ora. Come nel caso dei sensori idrometrici verrà richiesto al sistema di attivare il servizio di notifica eventi per quei sensori indicando però una lista di numeri di telefono a cui inviare un sms per informare della situazione di pericolo.

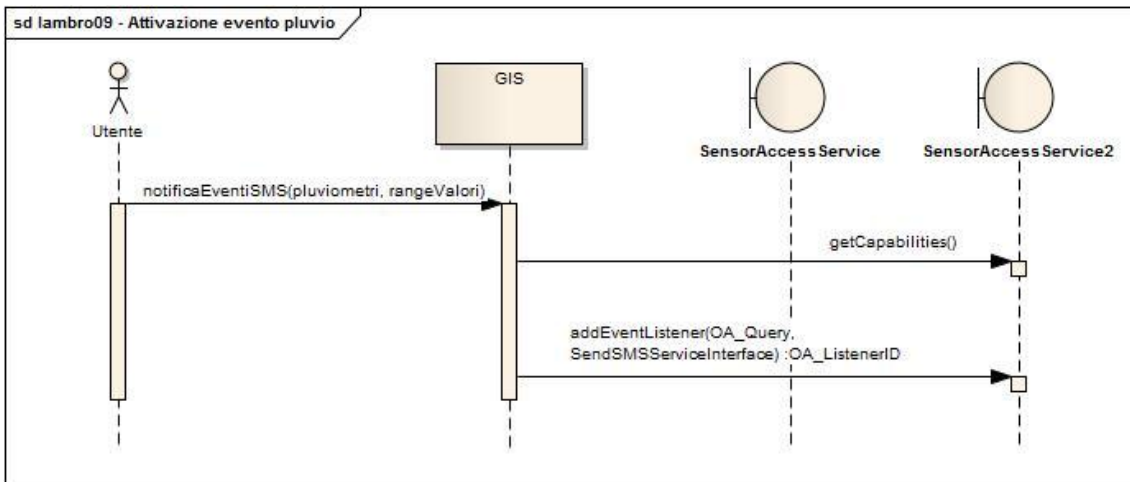


Figura 34 - Attivazione del servizio di notifica dei sensori

Dal grafico si può vedere come il sistema interroghi i sensori sulla possibilità di attivazione del servizio di notifica eventi. Avendo ricevuto una risposta affermativa richiama il servizio di notifica indicando come operazione di risposta il servizio SendSMSService il quale invierà ad una lista di numeri indicati un messaggio per avvertirli del superamento dei valori di allerta.

Ricerca Strutture a rischio ambientale

L'utente è riuscito dunque a recuperare tutte le informazioni idrogeologiche desiderate e prosegue quindi nella composizione dello scenario focalizzandosi sull'analisi del multirischio: egli cercherà quelle strutture che trattano materiali tossici e sono situate nell'area di studio in quanto ritenute come possibili entità scatenanti un ulteriore livello di emergenza (es. sversamento di liquidi inquinanti).

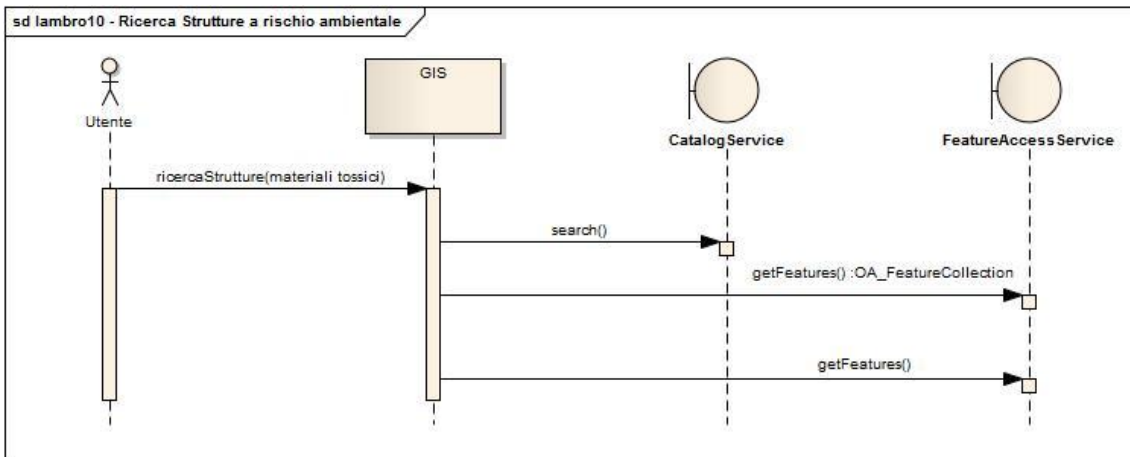


Figura 35 - Ricerca Strutture a rischio ambientale

Il sistema attraverso il servizio catalogo va a ricercare quei servizi di tipo *FeatureAccessService* che permettono l'accesso ai dati relativi alle installazioni industriali e alla loro funzione. Una volta recuperate l'elenco delle installazioni che corrispondono ai criteri di ricerca dei servizi il sistema, come nel caso dei sensori, mostra sulla mappa la posizione delle strutture che trattano materiali tossici e le informazioni riguardo a tipologia dei composti trattati, volume di materiali trattati, volumi stoccati, numero persone impiegate, ecc.

Attivazione del Servizio di Auto Aggiornamento dello Scenario

Nell'ottica di avere uno scenario sempre aggiornato l'utente richiede la notifica di eventuali modifiche all'elenco delle strutture di trattamento materiali tossici così come al numero delle risorse di emergenza come ambulanze o autopompe e l'aggiornamento giornaliero dei valori dei sensori presenti nello scenario.

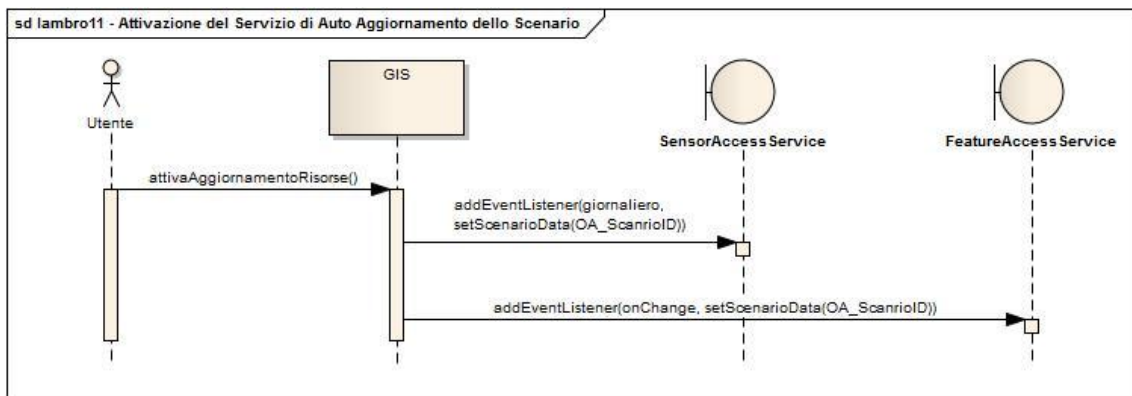


Figura 36 - Attivazione del Servizio di Auto Aggiornamento

Per implementare il concetto di auto aggiornamento dello scenario è ancora possibile utilizzare il concetto di evento: il sistema attiva la funzione *addEventListener* per ogni risorsa coinvolta e imputerà le regole di notifica in base alla tipologia di risorsa: richiederà la notifica di aggiornamento dell'elenco delle strutture solo se il loro numero cambia mentre per l'aggiornamento dei sensori effettuerà una richiesta di tipo temporale o basata sull'uscita da un range di valori ritenuti "modifiche non rilevanti". Per comunicare al servizio che effettua la notifica le modalità di aggiornamento della risorsa contenuta dello scenario il sistema indica *setScenarioData* come l'operazione da richiamare a fronte di un evento: così facendo ogni volta che il dato cambierà lo scenario sarà aggiornato e qualsiasi utente che accederà al servizio scenario per richiedere uno *screenshot* della situazione troverà i dati aggiornati.

L'aggiornamento dello scenario viene quindi effettuato in modo continuo e trasparente all'utente che ha creato e pubblicato lo scenario. E' possibile pensare che una nuova struttura che tratta materiali pericolosi venga attivata nella zona studiata: in questo caso sarà il servizio di raccolta dei dati delle strutture ad aggiornare lo scenario. Lo stesso vale per i valori letti dai sensori: ad intervalli di tempo stabiliti dall'utente che ha richiesto il servizio di aggiornamento il servizio di gestione dei sensori notifica il nuovo valore allo scenario al fine di avere dati sempre aggiornati.

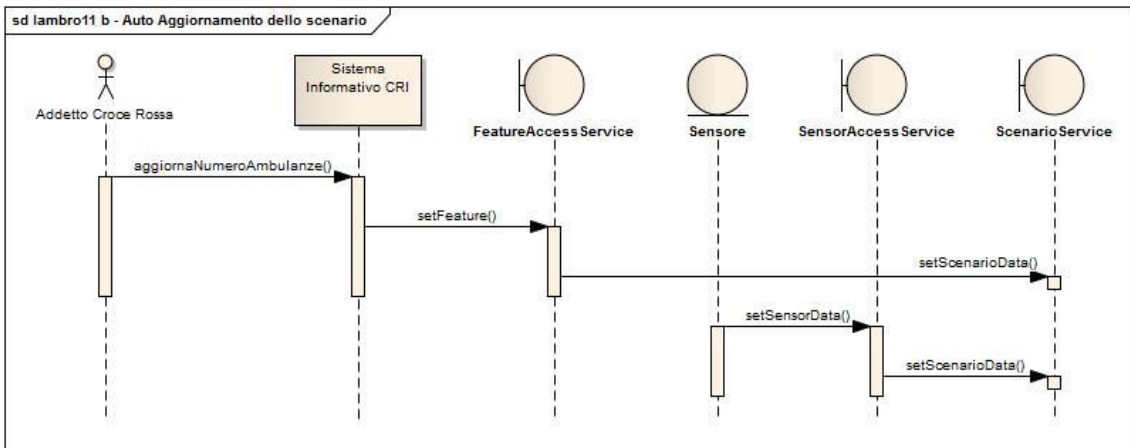


Figura 37 - Auto Aggiornamento dello scenario

Nel grafico riportato si vedono due tipi di aggiornamento:

- il primo è dovuto alla modifica da parte di un operatore della croce rossa del numero di ambulanze all'interno del sistema informativo dell'organizzazione. Questo sistema è connesso all'OSN e va a pubblicare nel *FeatureAccessService* i dati relativi appena modificati. Il FAS a fronte dell'aggiornamento verifica che l'entità modificata è "sotto osservazione" da parte di un qualche oggetto quindi richiama il metodo del servizio specificato in fase di attivazione del sistema di notifica. In questo caso il FAS richiama il metodo *SetScenarioData* che aggiornerà i dati presenti nello scenario.
- Il secondo è dovuto all'aggiornamento del valore letto da un sensore. In questo caso è sempre il *SensorAccessService* ad occuparsi di aggiornare i dati dello scenario attraverso il metodo. La frequenza e le modalità di aggiornamento sono state definite dall'utente in fase di set-up del servizio di notifica.

Ricerca Strutture Pronto intervento

L'utente per completare la valutazione dello scenario di rischio deve conoscere le strutture e i mezzi di pronto intervento presenti nell'area interessata e nelle zone circostanti. Egli è interessato a conoscere l'ubicazione delle stazioni dei pompieri e i relativi mezzi a disposizione, le sedi dei servizi di trasporto medico, la disponibilità dei mezzi e del personale disponibile per la gestione dell'emergenza. Anche queste

informazioni verranno visualizzate sulla mappa della zona ed eventualmente al di fuori di questa nel caso le risorse coinvolte fossero situate in zone limitrofe.

L'interazione tra sistema e rete di servizi effettua le operazioni di ricerca e recupero di informazioni reperibili da un *FeatureAccessService* come nel diagramma di Ricerca Strutture a rischio ambientale.

Ricerca documenti report dell'area interessata

Come ultima operazione prima di procedere alla valutazione dello scenario, l'utente ricerca nel sistema la presenza di eventuali documenti di analisi della zona in questione o di report storici. Il sistema effettua la ricerca e mostra all'utente la lista dei file trovati e la relativa descrizione. L'utente decide di consultare alcuni dei documenti presenti nella lista: in questo caso il GIS mostrerà a video il contenuto dei file trovati in base al tipo dichiarato nel file stesso: se il file ricevuto è un documento pdf questo avvierà l'applicazione di visualizzazione del programma, se questo fosse un filmato il sistema riprodurrà il file con un riproduttore video e così via.

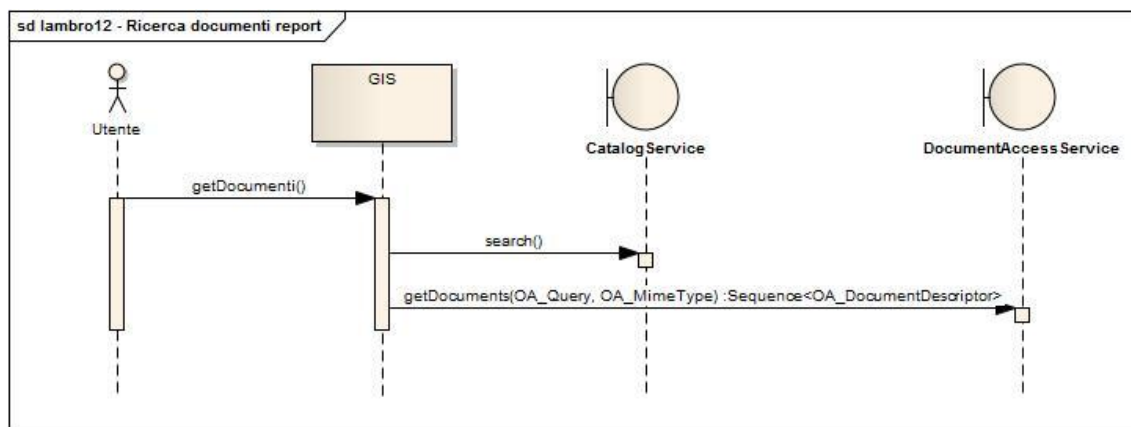


Figura 38 - Ricerca documenti report dell'area interessata

Si nota come il servizio di catalogo ricerchi un nuovo tipo di servizio, il *DocumentAccessService*, che permette il recupero di documenti di vario tipo. Una volta riconosciuto il linguaggio con cui interrogare il nuovo servizio il sistema richiede a questo tutti i documenti corrispondenti ai criteri impostati dall'utente quindi restituisce a quest'ultimo l'elenco dei documenti.

Aggiunta documento di analisi del rischio

L'utente, nelle fasi precedentemente descritte, ha composto per mezzo dell'integrazione tra il sistema GIS da lui utilizzato e una rete di servizi basata sull'architettura Orchestra, uno scenario di rischio vero e proprio sul quale effettuare una analisi del rischio e formulare un piano di intervento nel caso un evento idrogeologico si verificasse.

Il risultato finale dell'analisi è un documento che l'utente potrà o mantenere salvato internamente al proprio sistema o pubblicare in rete per l'accesso da parte di altri operatori del settore. A fronte di una richiesta di pubblicazione del documento il sistema si occuperà invocare il servizio di gestione documentale.

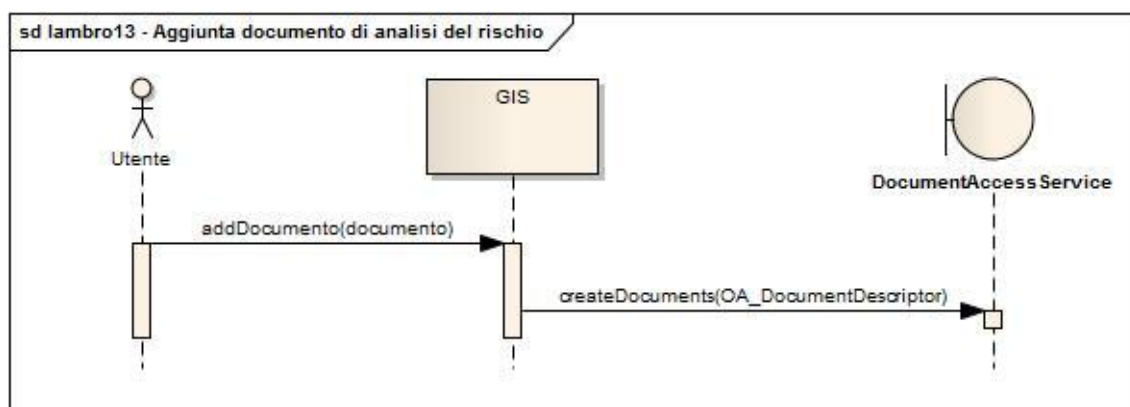


Figura 39 - Aggiunta documento di analisi del rischio

Il sistema per pubblicare il documento desiderato utilizza il *DocumentAccessService* precedentemente trovato per il recupero delle informazioni. Il metodo chiamato per la pubblicazione però è *createDocuments* il quale permette la creazione e il salvataggio del documento in oggetto, della sua descrizione e le sue meta informazioni.

Pubblicazione dello scenario

Oltre a pubblicare il documento di analisi del rischio l'utente vuole condividere lo scenario appena creato con il resto del network: attraverso la funzione di pubblicazione il sistema si occuperà di tutte le operazioni di salvataggio dello scenario nell'ambiente condiviso.

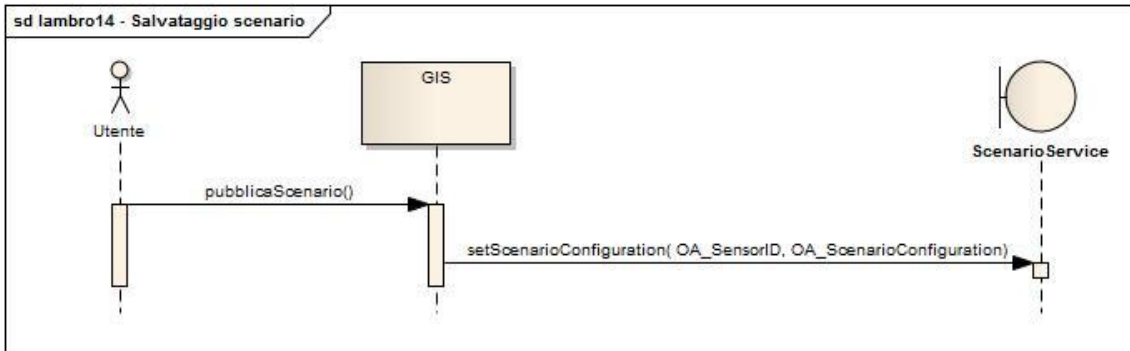


Figura 40 - Pubblicazione dello scenario

Il sistema utilizza le funzionalità dello *ScenarioService* già utilizzato all'inizio della composizione dello scenario e dal quale il sistema ha già ottenuto un identificativo univoco con il quale pubblicare lo scenario nel sistema senza andare incontro a problemi di duplicazione o di sovrapposizione tra scenari: il sistema chiama la funzione *setScenarioConfiguration* indicando l'id dello scenario e tutta la lista dei servizi che lo compongono. Al termine della procedura tutta la struttura dello scenario visto come servizio di servizi è stata salvata in modo persistente all'interno della base dati utilizzata dallo *ScenarioService* e tali informazioni saranno disponibili agli altri utenti che sono connessi al network.

Inserimento dati aggiornati nello scenario

Come ultima operazione l'utente vuole caricare nello scenario appena pubblicato tutti i dati relativi a sensori, risorse, documenti aggiornati: per fare questo il sistema implementerà la funzione pubblicazione dei dati relativi allo scenario

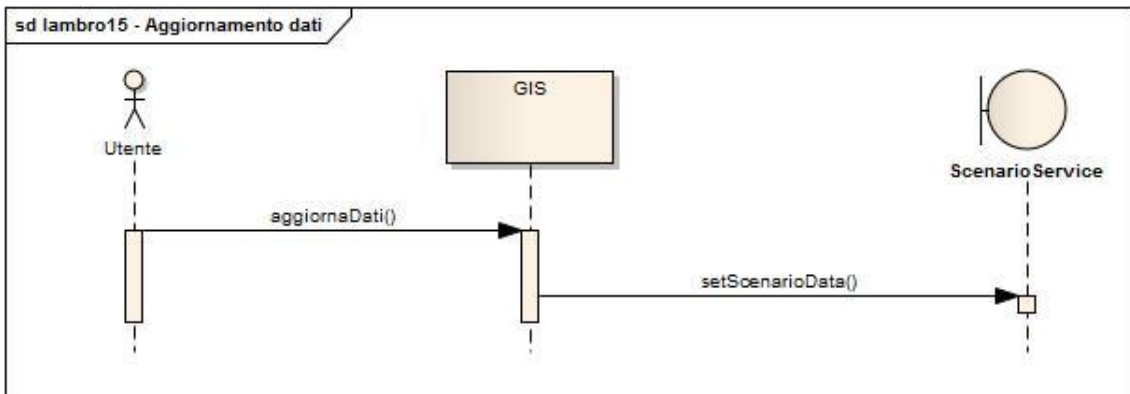


Figura 41 - Inserimento dati aggiornati nello scenario

Il sistema utilizzando sempre lo *ScenarioService* chiama il metodo *setScenarioData* il quale andrà ad aggiornare i valori relativi ad ogni *ScenarioObject* contenuto nello scenario in modo da pubblicare uno screenshot aggiornato della situazione analizzata

6.2.2 Valutazione della situazione Attuale di uno Scenario

Si prende in considerazione la situazione in cui l'utente del sistema vuole reperire e valutare la situazione di un determinato scenario; egli appartiene ad un ente esterno rispetto all'organizzazione che è responsabile della creazione degli scenari di rischio ed opera un applicativo che permette la visualizzazione dei dati geospaziali connesso alla rete di servizi OSN.

L'utente si connette al sistema e, dopo essersi autenticato, richiede al sistema il reperimento di uno scenario di rischio che interessa la zona attraversata dal fiume Lambro situata ad est di Milano attraverso lo strumento di ricerca messo a disposizione dall'applicazione utilizzata.

Il sistema recupera tutte le informazioni relative allo scenario e le visualizza sul monitor: l'utente avrà a disposizione una serie di informazioni dettagliate relative allo scenario che vanno dalla mappa della zona alla posizione dei sensori e delle infrastrutture civili presenti nella zona. L'utente può conoscere anche il valore dei sensori presenti nello scenario: tale valore sarà aggiornato in base alle politiche

applicate in fase di creazione dello scenario; se il creatore dello scenario ha attivato il servizio di notifica di eventi, al fine di aggiornare il dato dei sensori, essi indicheranno un valore “fresco”.

Nel caso l’utente desiderasse registrarsi alla notifica di eventi egli dovrà prima chiedere al sistema di scaricare la configurazione dell’intero scenario per poi andare ad attivare la richiesta di notifica di eventi.

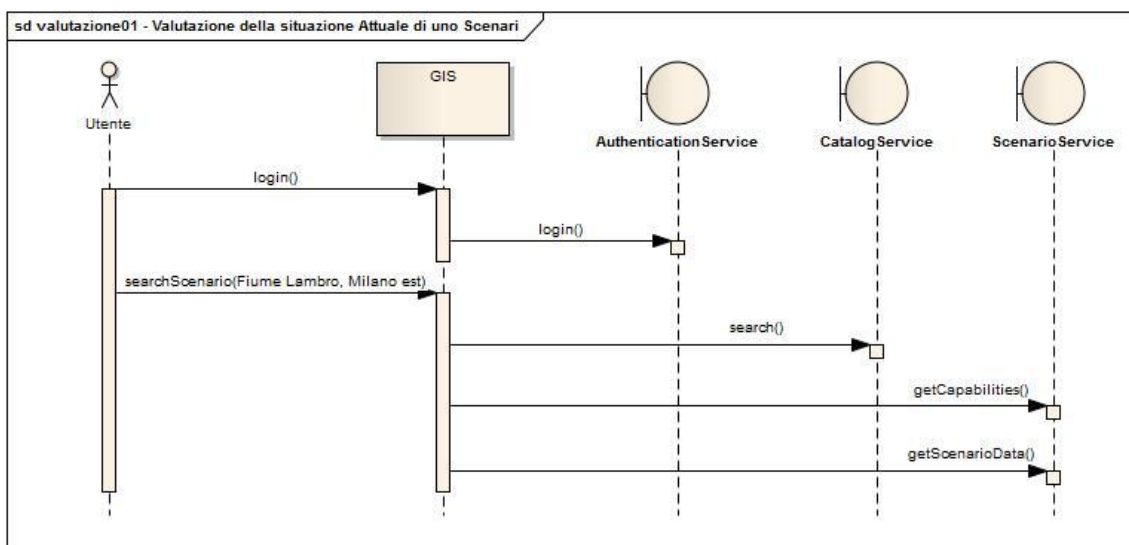


Figura 42 - Valutazione della situazione attuale dello scenario

Si osserva dal grafico l’interazione totale tra utente e sistema durante le fasi di reperimento dello *screenshot* dello scenario: dopo essersi autenticato il sistema attraverso il servizio catalogo recupera le informazioni per interrogare lo *ScenarioService*. Da quest’ultimo sarà possibile richiamare il metodo *getScenarioData* che restituirà i dati desiderati.

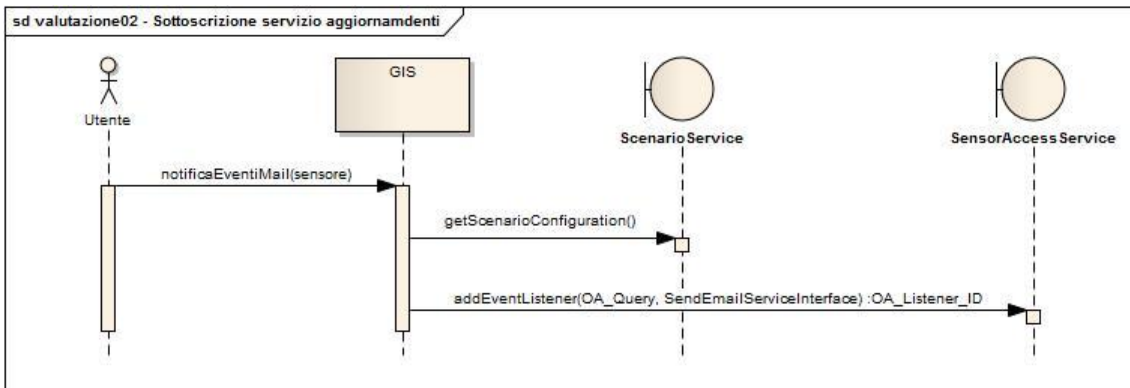


Figura 43 - Sottoscrizione del servizio aggiornamento

Nel caso l'utente volesse registrarsi alla notifica di eventi legati alla modifica dello stato di una risorsa presente nello scenario occorre che il sistema chiami il metodo *getScenarioConfiguration* per conoscere ogni singolo servizio che compone lo scenario

6.2.3 Ricerca di Dati Storici

A seguito di un evento piovoso di forte entità, la sala provinciale di PC desidera confrontare i dati in tempo reale con eventi passati, al fine di avere un supporto decisionale nello stabilire quali risorse attivare e quali provvedimenti prendere.

I dati in possesso della sala provinciale sono il livello delle acque rilevato tramite intervento umano presso il ponte tra Cologno Monzese fraz. San Maurizio e Sesto San Giovanni nei pressi dello svincolo autostradale, le previsioni in mm di pioggia ottimistiche e pessimistiche per le successive 48h.

Ricerca di scenari simili

L'operatore utilizza un sistema GIS collegato agli altri centri di protezione civile e collegato all' OSN da cui è possibile usufruire di diversi servizi collegati allo scenario di rischio.

L'operatore di PC ha a disposizione nel sistema lo scenario di rischio in tempo reale e visualizza sul monitor la mappa della zona interessata, l'ubicazione e i valori letti dai sensori idrometrici, l'ubicazione e il valore dei dati idrometrici rilevati da un altro

operatore (è possibile considerare la lettura manuale del dato come un “sensore umano”) e i dati delle previsioni meteo.

L’utente richiede al sistema di recuperare eventuali scenari passati. In prima istanza interroga il servizio ricercando situazioni simili alla situazione presente, grazie al concetto di similitudine tra scenari; l’utente ha richiesto esplicitamente al sistema di dare una priorità maggiore ai parametri relativi al livello del fiume Lambro durante il calcolo della similitudine tra i vari scenari.

Il servizio restituisce 2 screenshot di episodi simili: uno al termine di un evento piovoso, il secondo verificatosi durante l’evento piovoso stesso.

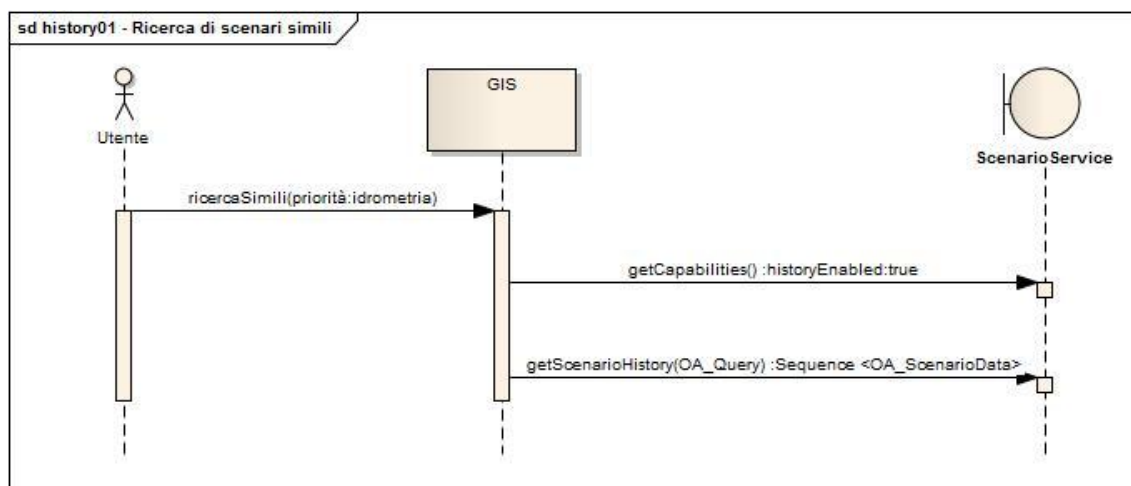


Figura 44 - Ricerca di scenari simili

Nel grafico si osserva l’intera interazione tra Utente – GIS – Servizio Scenari; si presuppone che l’utente, che si trova durante la fase di gestione dell’emergenza, sia già autenticato nel sistema e quest’ultimo sia già a conoscenza dell’end-point dello *ScenarioService* senza dover quindi ricercare il servizio attraverso il catalogo.

Il sistema richiama prima di tutto al servizio scenario il metodo *getCapabilities* per valutare se il servizio di history è abilitato. Il parametro booleano *historyEnabled* ricevuto contiene il valore “vero” quindi il GIS può procedere con la richiesta di scenari simili passati. Tale richiesta avviene attraverso il metodo *getScenarioHistory* che

restituisce una serie di screenshot di scenari (gli *ScenarioData*). Per identificare quali tipologie di scenario devono essere restituite il sistema traduce le richieste dell'utente relative all'individuazione geografica dello scenario e ai parametri per valutare la similitudine in un'opportuna *OA_Query*.

Recupero scenari passati

L'operatore dopo aver analizzato gli screenshot ricevuti dal sistema richiede gli scenari registrati nelle 48 ore successive all'ora in cui sono stati rilevati i due screenshot simili al fine di valutare lo sviluppo delle condizioni di rischio.

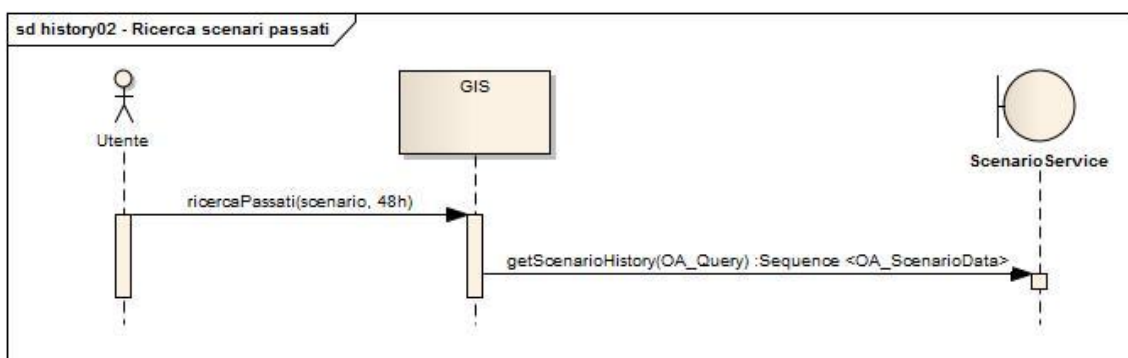


Figura 45 - Recupero scenari passati

La funzionalità di reperimento di scenari storici è supportata dallo *ScenarioService* ed è accessibile sempre dal metodo *getScenarioHistory*: l'unico parametro che cambierà rispetto alla richiesta precedente riguarda l'oggetto *OA_Query*; in questo caso *OA_Query* conterrà gli identificativi degli scenari già recuperati da cui si desidera partire nella valutazione dell'evoluzione storica.

6.3 Il caso Premana

Premana è un comune della provincia di Lecco situato tra la Valsassina e la Valtellina. All'interno del territorio comunale si è osservato un fenomeno di Deformazione Gravitativa Profonda di Versante ad opera del torrente Varrone il quale, con il suo processo di erosione determina uno scivolamento di materiale roccioso sedimentario e

non verso il fondovalle. Lo studio geologico effettuato in zona ha determinato come tale erosione aumenti in modo significativo durante episodi piovosi nei quali il torrente si ingrossa. Al fine di valutare e monitorare il rischio di eventuali frane (stabilite attraverso studi geologici e simulazioni), le autorità hanno installato alcuni sensori al fine di misurare e valutare al meglio la situazione. Come descritto dalla relazione, “dovendo inevitabilmente convivere con questo fenomeno, si dovrà provvedere a dotarsi degli strumenti di controllo e prevenzione necessari, che consentano di gestire al meglio eventuali, future condizioni di criticità”, è stato predisposto uno scenario di Rischio. (Savazzi, 2005)

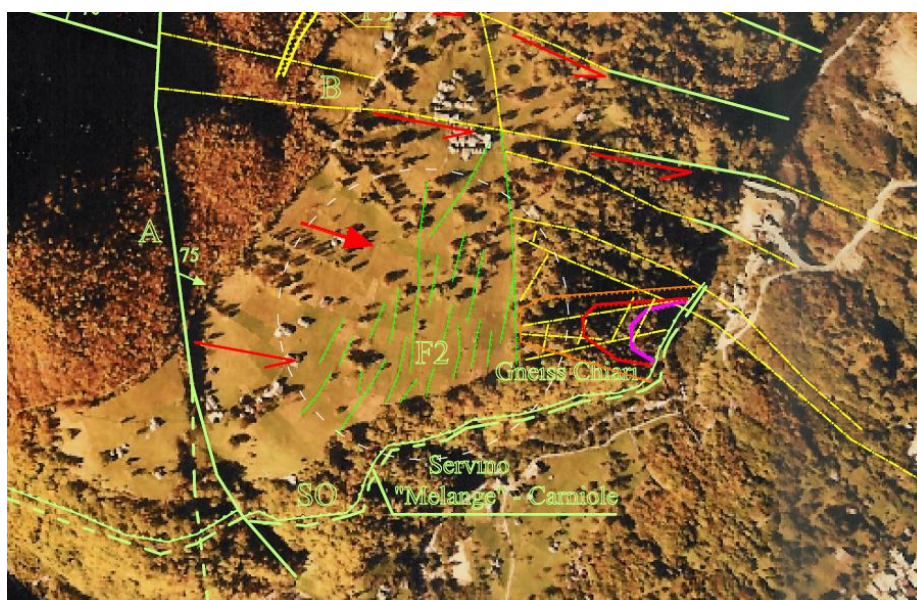


Figura 46 - Scenario di Rischio Cartaceo Premana

6.3.1 Notifica di Evento su scenario

L'utente che utilizza il sistema è un esperto del dominio, in questo caso è un geologo che sta valutando i rischi ambientali nella zona di Premana. Egli opera su un sistema che è stato interfacciato con una rete che si basa sull'architettura Orchestra estesa, dalla quale è quindi possibile usufruire dei servizi di accesso a scenari di rischio ed è possibile richiedere la notifica in merito a particolari eventi.

Ricerca Scenario di Premana

Come prima operazione l'utente attraverso il suo software di analisi vuole visualizzare uno scenario di rischio relativo all'area di studio: il sistema in modo trasparente si interfacerà al network dei servizi e cercherà un servizio di gestione degli scenari dal quale poter reperire la configurazione desiderata.

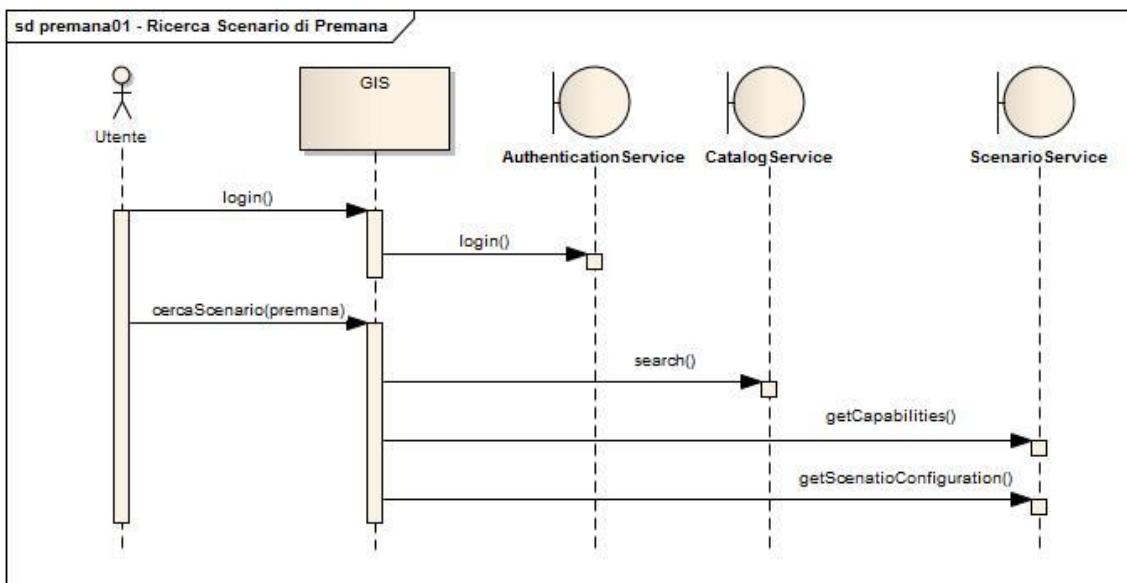


Figura 47 - Ricerca Scenario di Premana

Vediamo come anche questo sistema faccia uso del servizio catalogo per il reperimento di risorse che non conosce. Una volta ottenuto l'end-point del servizio di gestione degli scenari, lo *ScenarioService*, questo invoca il metodo *getScenarioConfiguration* grazie al quale è possibile ricostruire lo scenario in modo esatto ottenendo tutti i reference degli oggetti inseriti nello scenario.

Lettura configurazione e analisi dello scenario

Una volta ottenuta la configurazione il client può interrogare in cascata tutti gli oggetti componenti lo scenario al fine di ottenere il dato il quel momento

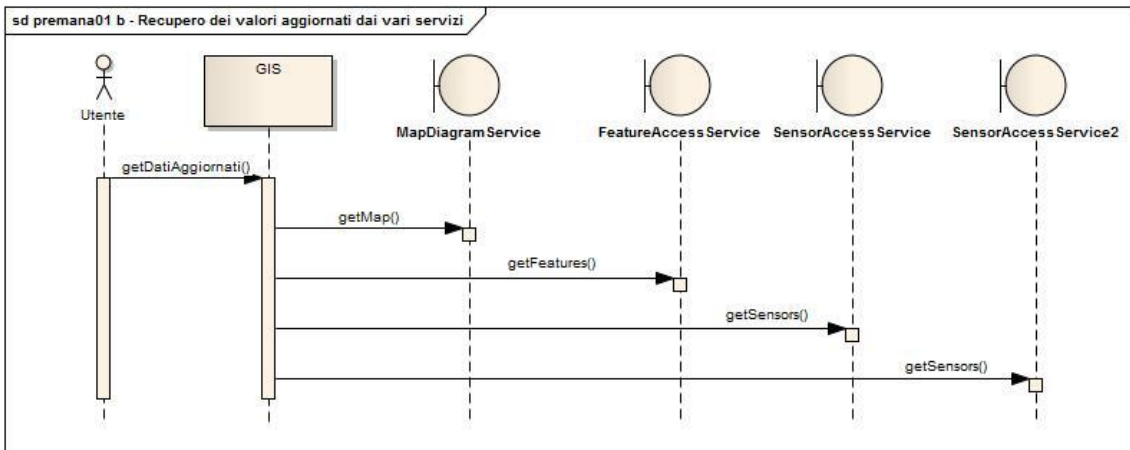


Figura 48 - Recupero dei dati a partire dalla configurazione di Scenario

Una volta completato il recupero dei dati, il sistema client riporta visivamente lo scenario di rischio della zona Premana precedentemente configurato e pubblicato da un altro utente esperto di dominio: l'utente del sistema visualizza la mappa della zona, i sensori disponibili nell'area, le infrastrutture civili presenti e l'ubicazione delle risorse di pronto intervento.

Grazie a queste informazioni l'esperto può avere una visione dello scenario completa ed aggiornata per una migliore valutazione del rischio.

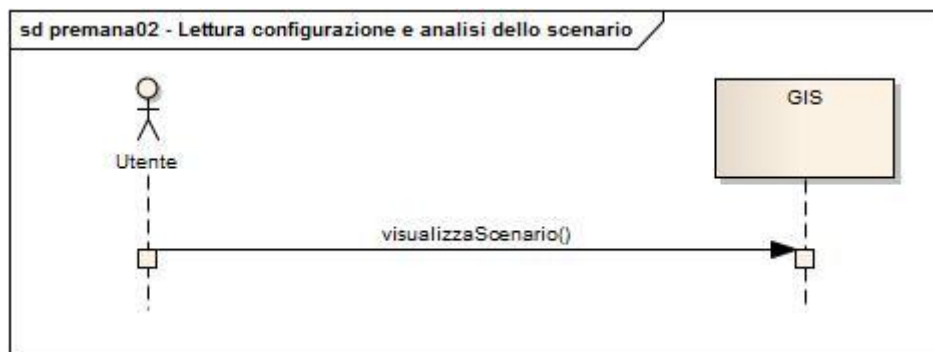


Figura 49 - Lettura configurazione e analisi dello scenario

In questa ultima fase l'utente interagirà solamente con il sistema in quanto tutte i dati di configurazione dello scenario e le informazioni necessarie per la visualizzazione di questo sono state già recuperate.

Attivazione servizio notifica eventi

Nell'analisi dello scenario considerato l'esperto reputa che le informazioni messe a disposizione da alcuni servizi presenti nello scenario sono particolarmente importanti e desidera quindi essere informato se i valori resi disponibili da tali servizi superassero contemporaneamente un determinato valore di soglia (ognuno secondo il proprio dominio); ad esempio nel caso in cui il servizio di previsione del tempo stimasse la caduta di x millimetri di pioggia per le 48 ore successe e che i sensori piezo elettrici rilevassero dei valori diversi da quelli di riferimento vorrebbe essere avvisato via e-mail di questa situazione. L'utente inserisce quindi nel sistema la richiesta e il servizio Scenario, una volta registrata la richiesta stessa, si occuperà gestire il controllo dei dati interessati e gestire l'eventuale notifica di evento.

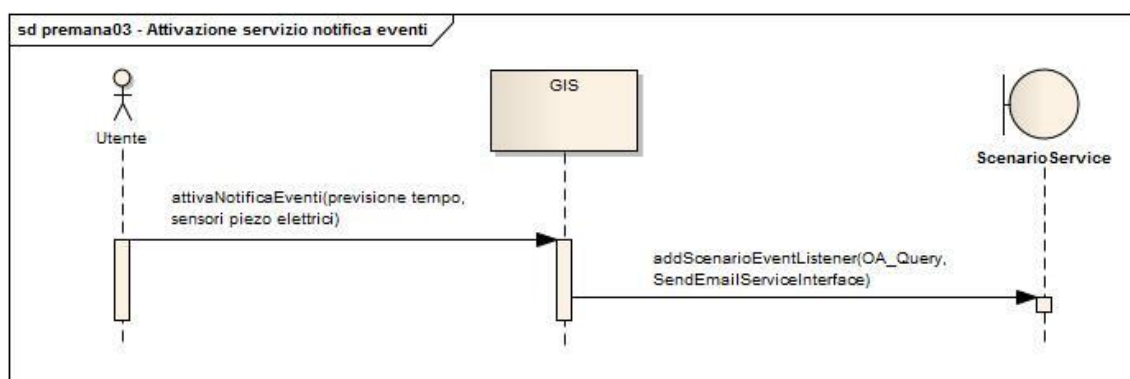


Figura 50 - Attivazione servizio notifica eventi

Il sistema attraverso l' OSN accede allo *ScenarioService* ed invoca il metodo *AddScenarioEventListener* il quale permette la configurazione di un listener associato ad un evento di modifica di alcune condizioni associate allo scenario. In questo caso il sistema oltre ad identificare gli oggetti sui quali effettuare il monitoraggio indica il servizio da chiamare nel caso l'evento si sollevasse: a fronte di una modifica il servizio chiamerà *SendMailService*.

Conclusioni

Con il presente documento si è voluto presentare una possibile estensione dell'architettura Orchestra al fine di implementare alcuni concetti ritenuti rilevanti al fine di rendere possibile un vero supporto informativo alla gestione dell'emergenza in ottica Augustus 2.0.

La scelta di utilizzare l'architettura Orchestra nasce dalla volontà di definire un'architettura indipendente dalle metodologie di comunicazione e di implementazione che, come affermato dallo stesso gruppo di lavoro di Orchestra, cambiano nel giro di una decade. La definizione risulta quindi volutamente astratta ed il più possibile indipendente dal dominio applicativo.

La prosecuzione di ulteriori ricerche nel campo dell'interoperabilità potrebbero definire una sintassi per la definizione dei concetti di scenario sia a livello di configurazione che di restituzione dei dati. Altro aspetto di approfondimento sicuramente interessante è la traduzione delle interfacce Orchestra in un'architettura che, sfruttando gli standard ad oggi esistenti e che verranno creati negli anni, possa fornire un supporto valido all'interoperabilità di EMIS.

Bibliografia

- Degiarde, E. (2008). *Sistema Integrato di Comunicazione per la Protezione Civile*.
Milano: IRER - Istituto Regionale di Ricerca della Lombardia.
- Forum Terremoto Centro Italia*. (s.d.). Tratto il giorno 11 21, 2010 da Corriere della Sera.it: http://forum.corriere.it/terremoto_nel_centro_italia/
- Frediani, C. (2010). La voce di Haiti su Twitter. *Corriere.it*.
- Galanti, E. (1997, Maggio-Giugno). Il "Metodo Augustus". *DPC Informa*.
- MapServer*. (s.d.). Tratto il giorno 10 08, 2010 da <http://mapserver.org/>
- Mike Botts, G. P. (2007, 12 28). OGC Sensor Web Enablement: Overview and High Level Architecture (OGC 07-165).
- Mogorovich P., M. P. (1988). *Automazione del Sistema Informativo territoriale. Elaborazione Automatica dei Dati Geografici*. Masson.
- Oasis Standard. (s.d.). Tratto il giorno 09 12, 2010 da Oasis Standard: <http://www.oasis-open.org/specs/>
- Oasis Standard. (s.d.). *Common Alerting Protocol Version 1.2*. Tratto il giorno 09 15, 2010 da <http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html>
- OGC. (2006). OpenGIS Web Map Service (WMS) Implementation Specification.
- Papazoglou, M. (2007). *Web Services: Principles and Technology*. Pearson-Education.
- Progetto OASIS - fp6*. (s.d.). Tratto il giorno 08 24, 2010 da <http://www.oasis-fp6.org/>
- Progetto Orchestra*. (s.d.). Tratto il giorno 11 19, 2010 da <http://www.eu-orchestra.org>

Provincia di Milano - Settore Protezione Civile. (s.d.). Tratto il giorno 10 21, 2010 da
http://www.provincia.milano.it/protezionecivile/emergenze/piano_emergenza_provinciale/index.html

Savazzi, G. (2005). *Comune di Premana: STUDIO GEOLOGICO DEL VERSANTE IN SINISTRA IDROGRAFICA DEL TORRENTE VARRONE.* Premana.

Warfield, C. (s.d.). *The Disaster Management Cycle.* Tratto da
http://www.gdrc.org/uem/disasters/1-dm_cycle.html