



POR CAMPANIA FESR 2007-2013

Asse 2	Obiettivo Operativo 2.1 <i>(Interventi su Aree Scientifiche di Rilevanza strategica)</i>	Obiettivo Operativo 2.2 <i>(Interventi di Potenziamento di sistema e di filiera della R&S)</i>
---------------	----------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------

BANDO PER LA CONCESSIONE DI AIUTI A PROGETTI DI RICERCA INDUSTRIALE E SVILUPPO SPERIMENTALE PER LA REALIZZAZIONE DI CAMPUS DELL'INNOVAZIONE

(indetto con Decreto Dirigenziale n° 327 del 11 Agosto 2009)



"Sistema Integrato di competenze per il MONitoraggio, la protezione ed il controllo delle infrastrutture idriche, fognarie ed Ambientali"

D8.1 Report progettazione del sistema di simulazione correlazione eventi e supporto alle decisioni del centro servizi

Relazione tecnica dell' OR 8 – "Progettazione del Dimostratore S.I.MON.A. su sito pilota (impianti e reti gestiti da G.O.R.I. S.p.A.)"	
SOGGETTO ATTUATORE: ENEA	CLASSIFICAZIONE DEL DOCUMENTO D.8.1/Rev.00/del/25/05/2015
RESPONSABILE SCIENTIFICO ing. Saverio De Vito	
RESPONSABILE DI OBIETTIVO REALIZZATIVO ing. Saverio De Vito	RESPONSABILE TECNICO OR ing. Saverio De Vito

Sommario

Sommario	2
1. Obiettivi e note preliminari	3
2. Architettura generalizzata	3
3. Tecnologie di base adottate	4
4. Definizioni Rilevanti	5
4.1 Sensoristica di riferimento: Generalità e Definizioni	5
4.2 Sensoristica di riferimento: Identificativi e caratterizzazione	6
4.3 Allarmi	7
5. Sistema di comunicazione	8
5.1 Architettura interna dell'Orchestratore	9
5.2 Modulo di interfacciamento	10
5.3 Simulation/Model Engine	10
5.3.1. Workflow Logic	14
6 Decision Support System	14
7 Simulatore Statico di Rete Fognaria	15
8 Simulatore Dinamico di Rete Fognaria	15
9 Simulatore di Consumo Energetico	15
Riferimenti	16

1. Obiettivi e note preliminari

Obiettivo del presente documento è presentare le informazioni e i diagrammi relativi alla progettazione del centro servizi del sistema SIMONA. Dato il contenuto fortemente tecnico è stata redatta una mini bibliografia di supporto per l'accesso alle informazioni riguardanti acronimi e tecnologie informatiche utilizzate. Per questo motivo, laddove ritenuto necessario, i termini tecnici rilevanti sono stati indicati e racchiusi tra parentesi quadre fungendo da chiave di entrata per la bibliografia raccolta nel capitolo 9 del presente documento. Per fini di leggibilità e maneggevolezza della documentazione sono state qui riportate alcune informazioni già riportate nel Deliverable 7.3.

2. Architettura generalizzata

Il sistema complessivo denominato SIMONA consta di differenti segmenti funzionali verticali interagenti. Il flusso di produzione ed elaborazione dati attraversa i livelli funzionali arricchendosi di valore semantico a partire dal dato raw, generato dal segmento sensoriale fino al segmento di interfaccia uomo macchina (vedi Fig.1). A meno dei sistemi di produzione del dato e di quelli atti alla visualizzazione del dato, i restanti livelli sono implementati in una struttura virtuale denominata Centro Servizi.

Il centro servizi si occupa dunque dell'integrazione, dell'immagazzinamento, del processamento ed infine della georeferenziazione del dato ai fini della visualizzazione.

L'integrazione del dato viene effettuata da sistemi appositi che permettono la ricezione ed il successivo invio ai sistemi di immagazzinamento di dati ontologicamente e tecnicamente molto eterogenei provenendo da reti di sensori con grandezze monitorate e tempi di campionamento fondamentalmente differenti.

Il successivo segmento si occupa dell'immagazzinamento del dato utilizzando come formato unificatore il paradigma NOSQL ed una base dati documentale (basata sulla tecnologia JSON nella sua variante BSON).

Un componente Orchestratore (localizzato nel Data Intelligence Layer) fa da tramite tra il Data Sink e lo HCI, implementando la "intelligenza" del sistema SIMONA, in termini di previsione (tramite simulatori) ed analisi (tramite il Decision Support System). In Figura 1.1 è possibile vedere la rappresentazione architetturale dell'intero sistema, il ruolo dello strato di Data Intelligence e l'Orchestratore in blu.

Infine un sistema di interfaccia uomo macchina georeferenzata si occupa dell'immagazzinamento del dato arricchito semanticamente ai fini specifici della visualizzazione avanzata su mappe.

In questo documento si analizzano specificamente gli aspetti tecnici di progetto delle componenti HW/SW del centro servizi in un'ottica particolare di integrazione funzionale. Viene altresì approfondito il sistema di orchestrazione del dato. Si rimanda a deliverable specifici la progettazione di dettaglio dei sistemi verticali ed in particolare i moduli HCI e di integrazione.

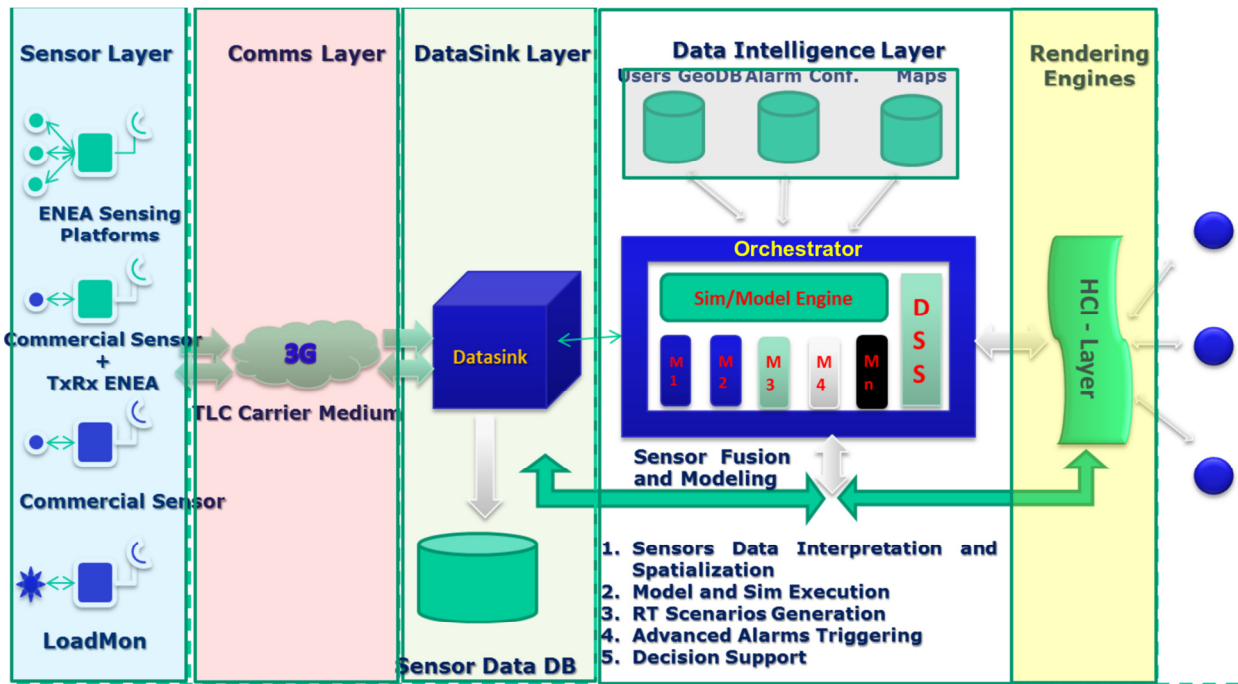


Figura 1 Architettura del sistema SIMONA

3. Tecnologie di base adottate

Per quanto riguarda il segmento di integrazione dati, ci si è basati su tecnologie Linux based e sistemi di computing su singola board. In particolare si è scelto di basarsi su sistemi Raspberry PI mod. B, board GSM e codice Java per la ricezione dati su rete GSM (messaggi SMS) e successivo invio verso i sistemi di storage. Il sistema operativo scelto è stato Raspbian. I sistemi di storage sensoriale sono basati su tecnologie NOSQL ed in particolare sul sistema MongoDB. Il formato di storage di quest'ultimo è documentale basato su oggetti BSON (JSON). Lo storage HCI è invece basato su tecnologie SQL ed in particolare PostgreSQL. Il componente Orchestratore della piattaforma SIMONA è stato progettato per integrare e coordinare moduli software di origine e natura fortemente eterogenee, supportandone l'aggiornamento o l'aggiunta nel ciclo di vita attesa del sistema. I modelli di simulazione da esso integrati e coordinati sono infatti realizzati come applicazioni indipendenti o come script per ambienti di calcolo scientifico quali Matlab, Octave, e R. Per rispondere a tali esigenze si è scelto di incentrare lo sviluppo per un sistema operativo Open Source (CentOS Linux), già utilizzato nell'infrastruttura CRESCO, ed di adottare Python [python] come linguaggio di programmazione principale, mentre per il *glue code* [2] ovvero per la gestione dei processi relativi a diversi moduli in esecuzione in uno stesso ambiente operativo e per l'adattamento dell'output da essi prodotto [3] si è scelto di utilizzare i linguaggi di scripting Bash [bash] e Awk [awk].

Per quanto riguarda Python, si sono utilizzate anche librerie per lo sviluppo di funzionalità accessorie, quali la gestione delle dipendenze di esecuzione dei diversi moduli software, la comunicazione con componenti esterne tramite API RESTful ed il parsing di documenti di testo.

Nello specifico, per la gestione delle dipendenze di esecuzione dei moduli e per la loro sincronizzazione, il componente Orchestratore include un workflow engine programmabile (modulo Simulation/Model Engine) progettato adottando il paradigma del flow-based programming [leaf], supportato da Python attraverso diverse librerie (ad es. PyF [pyf], PyLeaf [pyleaf], etc). I modelli di simulazione ed il Decision Support System, di cui il workflow engine coordina l'esecuzione, sono implementati utilizzando diverse tecnologie.

Per i modelli di simulazione del consumo energetico, si stanno sviluppando programmi ad-hoc parzialmente basati sull'ambiente di calcolo scientifico Matlab/Octave.

Per i modelli di simulazione della rete fognaria si stanno sviluppando programmi ad hoc in linguaggio C con le metodologie MPI per la parallelizzazione del calcolo e l'utilizzo di risorse HPC. Il processo di simulazione, che utilizza tecniche di ottimizzazione e algoritmi genetici, viene eseguito con l'ausilio dei codici SWMM5 [swmm] e GALib [galib], entrambi Open Source e scritti in C/C++. Parallelamente, si sta eseguendo un lavoro di porting di tali codici nell'infrastruttura di calcolo CRESCO di ENEA. L'esecuzione parallela del codice viene effettuata tramite librerie collegate al codice di calcolo e funzionalità messe a disposizione da CRESCO in modo trasparente al componente Orchestratore, che ne beneficia pur non gestendole direttamente.

4. Definizioni Rilevanti

4.1 Sensoristica di riferimento: Generalità e Definizioni

Il sistema di monitoraggio SIMONA ed in particolare la sua catena di elaborazione del dato utilizza una serie di definizioni rilevanti che assumono significato comune per tutti i componenti dell'architettura. Di particolare rilevanza sono le definizioni dei nodi di rilevazione che possono essere suddivisi in via primaria in sensori reali e sensori virtuali.

Con il termine sensore reale si fa riferimento ad un nodo sensoriale "fisico" dotato di un array di sensori fisico-chimici interagenti con il mondo reale ed atti alla rilevazione di grandezze fisico-chimiche attuali o derivate (proxy).

Per grandezze attuali s'intendono grandezze fisico-chimiche per cui esiste ed è installato un nodo sensoriale capace di una rilevazione diretta, ad esempio un sensore di pH che è capace di stimare il valore di pH della soluzione in cui è immerso. Per grandezze derivate s'intendono quelle per cui non esiste un sistema in grado di effettuare una rilevazione diretta (o almeno non è disponibile in Simona) pertanto se ne effettua una rilevazione indiretta mediante l'ottenimento di parametri surrogati che possano permettere mediante una opportuna legge di calibrazione una stima più o meno attendibile della grandezza originaria (es.: stima dei solidi sospesi mediante riflettanza/assorbanza nell'infrarosso, cfr. Sensore Loadmon). Altro esempio di questa tipologia sensoristica è il multisensore ENEA che rileva misure fisico chimiche dirette che vengono utilizzate per generare un'anomalia generica chimico-fisica nell'acqua.

Con il termine sensore virtuale si identificano informazioni di grado semantico elevato non direttamente rilevate da un sensore reale ma generate mediante una pre-elaborazione del dato proveniente da sensori reali mediante modelli elaborativi o simulativi. Un esempio di tale tipologia di sensori è la generazione di allarmi da parte del sistema DSS (sulla base di regole o algoritmi). In tal caso il sistema DSS genera una lettura di un sensore "virtuale" bistabile (0: Normalità, 1: Allarme) da interpretarsi come criticità. In generale tali allarmi possono essere generati quando occorresse un superamento di un singolo parametro specifico o il verificarsi simultaneo di condizioni e valori "critici" di più grandezze fisico-chimiche. Infine, l'ultima tipologia di sensori virtuali presente in SIMONA raggruppa i valori di grandezze fisico-chimiche in alcuni punti specifici della rete così come predetti e generati dai simulatori funzionali della rete, negli istanti di tempo definiti dall'orizzonte temporale di simulazione.

Per una schematizzazione di dettaglio si faccia riferimento alla seguente tabella:

Nodo Sensoriale	Tipologia	Carattere	Note
<i>Centralina multisensoriale commerciale</i>	Reale	Diretta	Valori generati da un array di sensori chimico-fisici a contatto
<i>Multisensore Loadmon</i>	Reale	Derivata	
<i>Centralina Multisensoriale ENEA</i>	Reale	Derivata	Genera una lettura di anomalia rispetto a condizione di normalità apprese durante il funzionamento relativamente al valore di sensori fisico chimici non a contatto
<i>Sensore di Presenza/Assenza</i>	Reale	Diretta	Presenza di liquido in condotta
<i>Sensori di grandezza fisico/chimica simulati</i>	Virtuali	Diretta	Rilevazioni in punti specifici dell' infrastruttura in un tempo futuro generati tramite simulazione predittiva
<i>Anomalie di regole</i>	Virtuali	Derivata	Anomalie generate sulla base di opportune regole che tengono conto dei valori simulati e/o direttamente rilevati da più nodi simultaneamente

4.2 Sensoristica di riferimento: Identificativi e caratterizzazione

I sistemi sensoriali sviluppati ed installati nel progetto SIMONA sono identificati singolarmente da un codice valido per l' intera catena di elaborazione del dato. La loro identificazione è completata dalla loro localizzazione specifica che viene utilizzata dai sistemi HCI per la visualizzazione del dato. Le tabelle seguenti mostrano i sistemi utilizzati, il loro identificativo, le coordinate geografiche e le variabili di riferimento.

Punti di installazione SONDE COMMERCIALI (rif. Schemi allegati alla richiesta di offerta)

ID di progetto SIMONA (D8.11)	Depuratore di Massa Centro	SOTTOCODICI CM1	Tipologia Sonda	GB_X	GB_Y	Precisione
1	Ingresso depuratore (portata)	CM1.1	Sonda di Misura Portata	2464273.3681	4496058.7735	5m
2	Ingresso depuratore (livello)	CM1.2	Sonda di Misura Livello	2464273.3681	4496058.7735	5m
3	Sonda multiparametrica commerciale (Loadmon)	CM1.3	Sonda Multiparametrica	2464276.0693	4496056.1863	5m
4	Pluviometro (ipotesi : installazione su palazzina uffici)	CM1.4	Pluviometro	2464265.5045	4496081.4549	5m

ID di progetto SIMONA (D8.11)	Sollevamento Torca	SOTTOCODICI CM2	Tipologia Sonda	GB_X	GB_Y	Precisione
1	Ingresso a gravità	CM2.1	Sonda Multiparametrica	2468173.3030	4494312.9269	5m
2	Pluviometro (ipotesi : installazione su locale fronte ufficio)	CM2.2	Pluviometro	2468201.8179	4494329.3786	5m

ID di progetto SIMONA (D8.11)	Sollevamento Gesigione	SOTTOCODICI CM3	Tipologia Sonda	GB_X	GB_Y	Precisione
1	Sonda multiparametrica in vasca interrata	CM3	Sonda Multiparametrica	2463504.8254	4494729.1090	submetrica

ID di progetto SIMONA (D8.11)	Marina della Lobra Alto	SOTTOCODICI CM4-CM5	Tipologia Sonda	GB_X	GB_Y	Precisione
1	Sonda multiparametrica in vasca interrata	CM4	Sonda Multiparametrica	2463798.5140	4495592.9243	5m
2	Pozzetto (misura Portata)	CM5.1	Sonda di Misura Portata	2463814.4495	4495573.9861	5m
3	Pozzetto (misura Livello)	CM5.2	Sonda di Misura Livello	2463814.4495	4495573.9861	5m

Le variabili di riferimento dei sensori commerciali

ID (8.11)	Sito	Sonda	Parametri	Unità di misura	Range
CM1	Ingresso Depuratore Massa Centro	NIVUS OCM-FM	Portata	l/s	(in base alle impostazioni e secondo le dimensioni del canale di installazione)
		NIVUS KDO-KP	Livello	cm	0-200
	Layout Loadmon (canale)	Sonda Condu Lyser II pro	temperatura	°C	(-5-75)
			Conducibilità	mS/cm	0-2000
		Ammo Lyser II pro	Ammoniaci (NH4)	ppm (mg/l)	0-1000
			PH	-	0 - 14
	Su copertura Palazzina uffici	Carbo Lyser	COD	mg/l	0-3750
			Solidi Sospesi (TSS)	mg/l	0-2500
	Loadmon	HD2013R	Pluviometro	Impulso	Fornisce un impulso ogni 0,1mm di pioggia
			COD SST		
CM2	Ingresso Sollevamento Torca	Sonda Condu Lyser II pro	temperatura	°C	(-5-75)
			Conducibilità	mS/cm	0-2000
		Ammo Lyser II pro	Ammoniaci (NH4)	ppm	0-1000
			PH	-	0 - 14
	Su copertura manufatto edificio fronteggiante l'ufficio	Carbo Lyser	COD	mg/l	0-3750
			Solidi Sospesi (TSS)	mg/l	0-2500
		HD2013R	Pluviometro	Impulso	Fornisce un impulso ogni 0,1mm di pioggia
CM3	Sollevamento Gesiglione	Sonda Condu Lyser II pro	temperatura	°C	(-5-75)
			Conducibilità	mS/cm	0-2000
		Ammo Lyser II pro	Ammoniaci (NH4)	ppm	0-1000
			PH	-	0 - 14
		Carbo Lyser	COD	mg/l	0-3750
			Solidi Sospesi (TSS)	mg/l	0-2500
CM4	Sollevamento Marina della Lobra	Sonda Condu Lyser II pro	temperatura	°C	(-5-75)
			Conducibilità	mS/cm	0-2000
		Ammo Lyser II pro	Ammoniaci (NH4)	ppm	0-1000
			PH	-	0 - 14
		Carbo Lyser	COD	mg/l	0-3750
			Solidi Sospesi (TSS)	mg/l	0-2500
CM5	Pozzetto Marina Lobra	NIVUS OCM-FM	Portata	l/s	(in base alle impostazioni e secondo le dimensioni del canale di installazione)
		NIVUS KDO-KP	Livello	cm	0-200

SENSORI PROTOTIPALI DI RILEVAZIONE : TIPOLOGIA "NON A CONTATTO"

ID di progetto SIMONA (D8.11)	Depuratore di Massa Centro	SOTTOCODICI CM1	Tipologia Sonda	GB_X	GB_Y	Precisione	
1	CM1	Loadmon	CM1.5	Sensore Loadmon	2464276.0693	4496056.1863	5m

ID di progetto SIMONA (D8.11)_ID		Tombino (TB)	GB_X	GB_Y	Precisione	ubicazione	Offering	Observation
1	PM1	TB0126	2464625,0559	4495747,2831	submetrica	via Rachine	Proto_Multisensore	Anomalia*
2	PM2	TB0003	2465256,0087	4495739,9446	submetrica	via San Francesco da Paola	Proto_Multisensore	Anomalia*
3	PM3	TB0057	2464405,7757	4495530,5123	submetrica	via Rivo a Casa	Proto_Multisensore	Anomalia*
4	PM4	TB0732	2463834,3793	4495381,3899	submetrica	via Fontanelle	Proto_Multisensore	Anomalia*
5	PM5	TB0312	2463332,2164	4494659,1460	submetrica	via Nastro d'Oro	Proto_Multisensore	Anomalia*
6	PM6	TB1383	246507,9994	4495211,5767	submetrica	via Reola	Proto_Multisensore	Anomalia*
7	PM7	TB1620	2467091,6686	4495164,9924	submetrica	C.so Sant'Agata	Proto_Multisensore	Anomalia*
8	PM8	TB1676	2468094,4940	4494369,5574	submetrica	Via Dei Galli	Proto_Multisensore	Anomalia*

SENSORI PROTOTIPALI DI RILEVAZIONE PRESENZA/ASSENZA REFLUI : TIPOLOGIA "A CONTATTO"

ID di progetto SIMONA (D8.11)		Tombino (TB)	GB_X	GB_Y	Precisione	Tipologia di punto monitorato	ubicazione	Offering	Observation
1	PPA1	TB0060	2464309.2289	4495541.8727	submetrica	Scaricatore di piena C30SCP04	via rivo casa	Sensore P/A	Presenza acqua
2	PPA2	TB0744	2463757.5808	4495569.7552	submetrica	Scaricatore di piena C30SCP11	via fontanella	Sensore P/A	Presenza acqua
3	PPA3	TB1501	2467065.6786	4494754.0909	submetrica	Scaricatore di piena C30SCP17	traversa III pigna	Sensore P/A	Presenza acqua
4	PPA4	TB1675	2468072.0114	4494417.0244	submetrica	Scaricatore di piena C30SCP14	via dei galli	Sensore P/A	Presenza acqua

* Anomalia (Livello, Concentrazione H2S, Rumore, T Liquido, T Ambientale, RH)

4.3 Allarmi

Il sistema SIMONA identifica diverse tipologie di allarmi in diversi momenti della catena di elaborazione del dato e a partire da differenti sottosistemi:

- Una prima e più intuitiva categoria è generata dal sistema HCI sulla base del valore raggiunto dalle letture di un singolo trasduttore. La rilevazione di tali allarmi viene effettuata al termine della catena di elaborazione ed è basata su un livello di soglia prefissato relativo alla singola grandezza rilevata da un singolo sensore (reale o virtuale) (ed alla localizzazione del sensore?) indipendente

dalle condizioni dell'intero sistema da monitorare. Un esempio può essere un superamento di una soglia sul livello di Solidi sospesi in un singolo sensore o la condizione di rilevazione di liquido da parte di un nodo sensoriale di presenza/assenza.

- Una seconda categoria di allarmi, se vogliamo più sofisticata, viene generata dal sistema di gestione degli allarmi sulla base di regole predeterminate ed algoritmi appositi che possono dipendere anche dalle condizioni del sistema globale e, generalmente, coinvolgono il valore letto da uno o più sistemi di rilevazione. Un esempio può essere la determinazione di una condizione di allarme "generale" attivata dalla simultanea identificazione di una presenza di liquido in tempo asciutto (rilevazione pluviometrica) da parte di uno dei nodi presenza assenza.
- Infine la terza categoria di allarmi è precipua dei Multisensori ENEA i quali generano un allarme relativo ad una anomalia rispetto ad una condizione di normalità generalmente rilevata da quello specifico nodo sensoriale, in quella specifica localizzazione in quella specifica ora del giorno. Questi nodi saranno infatti equipaggiati con un algoritmo di anomaly detection che, operando a bordo, sarà in grado di identificare una condizione di normalità imparandola sul campo e di rilevare eventuali anomalie locali.

Maggiori dettagli sono contenuti nel deliverable 8.4 Modulo Gestione Allarmi.

5. Sistema di comunicazione

Il dispiegamento fisico previsto per il componente Orchestratore è descritto nel Deployment Diagram in Figura 2, dal quale si evince che le comunicazioni avvengono sia con nodi remoti, sia tra moduli software in esecuzione sullo stesso nodo.

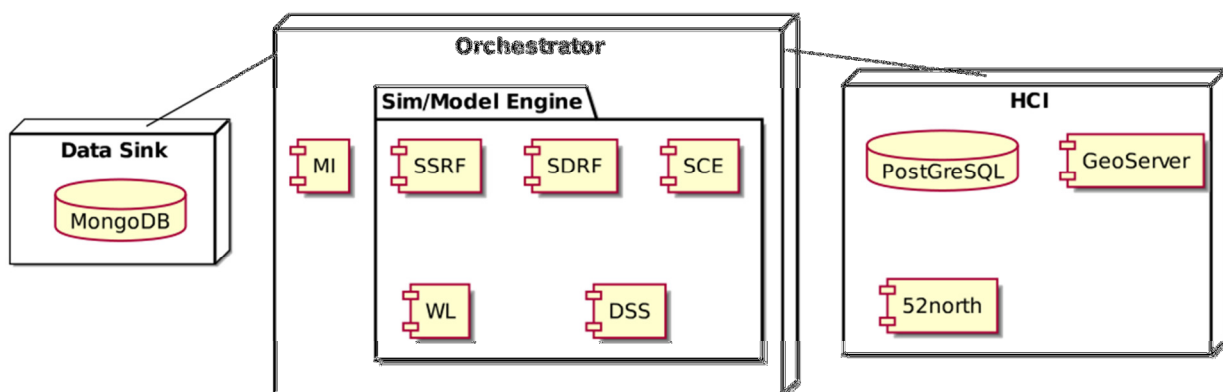


Figura 2: Deployment Diagram dell'Orchestratore

L'Orchestratore comunica remotamente con altri due componenti della piattaforma SIMONA: il Data Sink e lo HCI. Tali comunicazioni vengono effettuate tramite protocolli applicativi incapsulati in HTTP e trasportati su uno stack protocollare TCP/IP. I livelli fisico (PHY) e di accesso al canale (MAC), ai fini delle necessità prestazionali dell'architettura, possono essere forniti da collegamento LAN Ethernet e/o tramite xDSL. Le comunicazioni tra i moduli interni all'Orchestratore avvengono localmente all'ambiente di esecuzione. Nel caso di moduli implementati come applicazioni indipendenti, le comunicazioni avvengono utilizzando lettura / scrittura coordinata di file e pipe Unix; lo scambio di messaggi è sincrono, ed il parallelismo è gestito dal Simulation/Modeling Engine, che può eseguire parallelamente task che non hanno dipendenze

funzionali. Nel caso di componenti implementate come moduli Python, lo scambio di messaggi è sincrono ed avviene tramite chiamate a funzioni (metodi dell'interfaccia del componente) ed accesso a strutture dati condivise.

Alcuni moduli interni all'Orchestratore (es. i simulatori) possono effettuare elaborazioni di calcolo parallelo utilizzando la piattaforma CRESCO. Le comunicazioni relative alle funzionalità di calcolo parallelo non sono gestite esplicitamente dall'Orchestratore e sono trasparenti al livello di analisi considerato in questo documento, pertanto non saranno rappresentate nei Sequence Diagram e non saranno descritte nel workflow.

5.1 Architettura interna dell'Orchestratore

L'Orchestratore è un componente fortemente modulare, e comprende diverse sotto-componenti (denominate moduli nel seguito) rappresentate graficamente in Figura 2.1. Si distinguono il modulo Simulation/Model Engine (SME), responsabile della gestione dei modelli di simulazione (denominati SSRF, SDRF, SCE, ed in seguito specificati), il Decision Support System (DSS) ed il Modulo di Interfacciamento (MI), responsabile delle interazioni con componenti esterne all'Orchestratore.

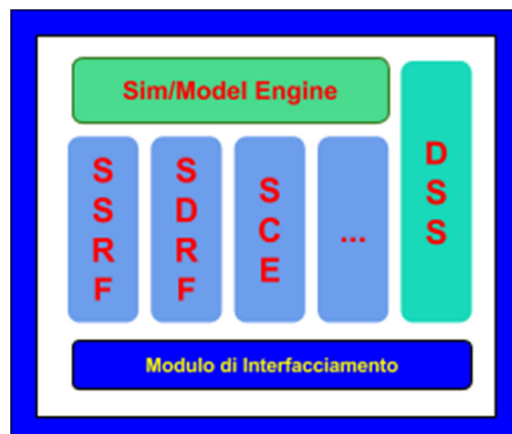


Figura 3 Architettura interna dell'Orchestratore

Per una visione d'insieme del progetto dell'Orchestratore si fa riferimento al Component Diagram in Figura 3 in cui l'Orchestratore è modellato utilizzando il tipico formalismo UML.

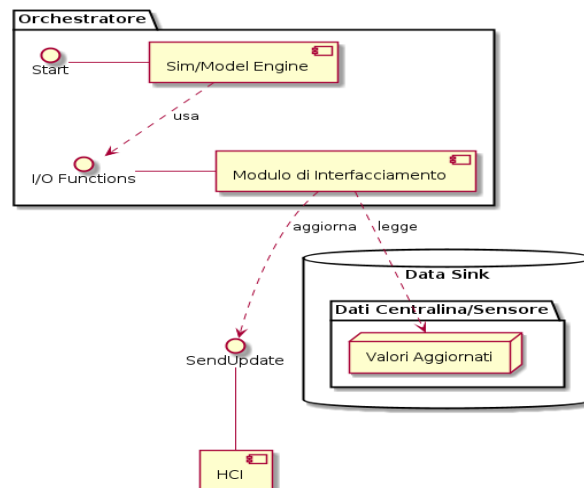


Figura 4: Component Diagram dell'Orchestratore

Come risulta evidente dal diagramma in figura 4, il sistema di simulazione utilizza interfacce fornite dal modulo di interfacciamento per l'ottenimento di informazioni aggiornate provenienti dal modulo di back-end qui denominato DataSink realizzato in tecnologia MongoDB. Nei seguenti paragrafi si descrive a livello architetturale la progettazione dei moduli interni al componente Orchestratore.

5.2 Modulo di interfacciamento

Il Modulo di Interfacciamento (MI) offre al Simulation/Model Engine (SME) tutte le funzioni di lettura (es. dal Data Sink) e scrittura degli aggiornamenti (es. verso lo HCI). La sua funzione è quella di disaccoppiare le fonti di dati dal funzionamento del sistema di federazione di simulatori. In questo modo, in ossequio ai principi di modularità, nuove fonti e nuovi utilizzatori di dati possono essere facilmente aggiunti al sistema riducendo la necessità di sviluppare nuovo software. Laddove necessario, in questo modulo è possibile implementare una cache locale che memorizzi i risultati delle ultime letture e, in caso di richieste agli stessi dati ripetute in un determinato intervallo temporale, restituisca direttamente i dati memorizzati in locale. In tal modo si può limitare l'accesso al Data Sink e quindi contenere la latenza dovuta dalle trasmissioni su rete ethernet. E' inoltre possibile aggregare gli aggiornamenti dello stato corrente e simulato dell'infrastruttura, limitando l'overhead di comunicazione e quindi l'uso di risorse trasmissive. Il Modulo di Interfacciamento sarà implementato in Python. Più nello specifico, per comunicare con il Data Sink sono state scelte le librerie PyMongo [pymongo], poiché nell'insieme di alternative disponibili (incluso generazione e parsing di risorse JSON veicolate tramite HTTP) è risultata una soluzione matura, robusta e considerata preferenziale dalla comunità di sviluppo Python per l'accesso al DBMS adottato per il Data Sink (MongoDB). Per comunicare con lo HCI layer sono state scelte le librerie Requests [requests] che faranno da wrapper di alto livello per le comunicazioni incapsulate in HTTP con uso del paradigma RESTful.

5.3 Simulation/Model Engine

Il Simulation/Model Engine (SME) è il componente effettivamente responsabile dell'orchestrazione dei simulatori e del Decision Support System. Esso implementa tutte le funzionalità dell'Orchestratore ad eccezione delle funzionalità di input-output da e verso i componenti esterni, che sono implementate dal MI come descritto precedentemente. Il SME è composto da simulatori eterogenei, modellati come realizzazioni del componente astratto *:Simulatore*, dal Decision Support System, che processa i risultati di simulazione, e dalla Workflow Engine Logic, che coordina tutti questi moduli. Si rimanda al Component

Diagram in Figura 5 per una visione d'insieme di questo componente. Le comunicazioni tra il SME ed il resto del sistema sono rappresentate nel Sequence Diagram in Figura 6, mentre una schematizzazione della logica di funzionamento del SME è riportata nell'Activity Diagram in Figura 7. Di ciascuno dei moduli inclusi nel SME viene riportata nel seguito la descrizione dettagliata.

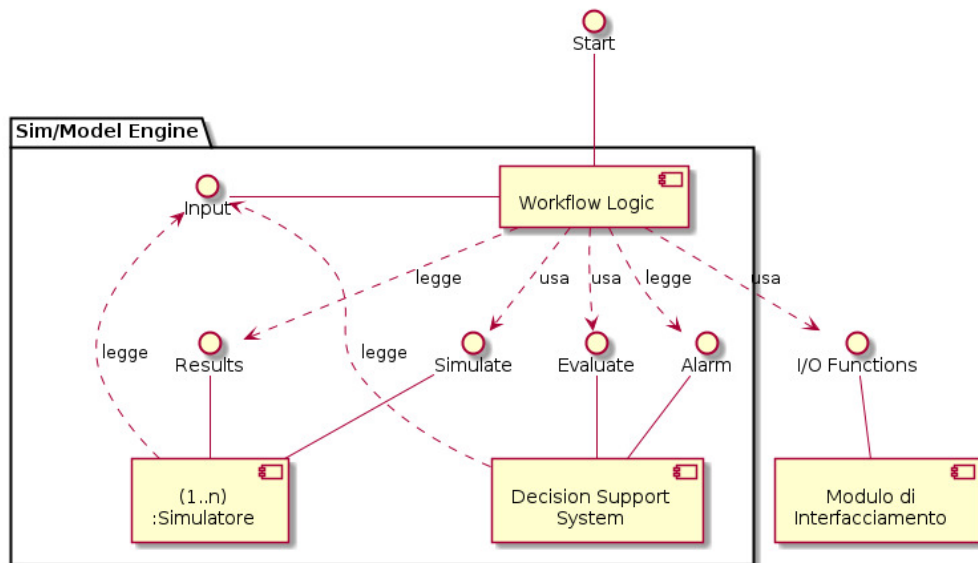


Figura 5: Component Diagram del Simulation/Model Engine

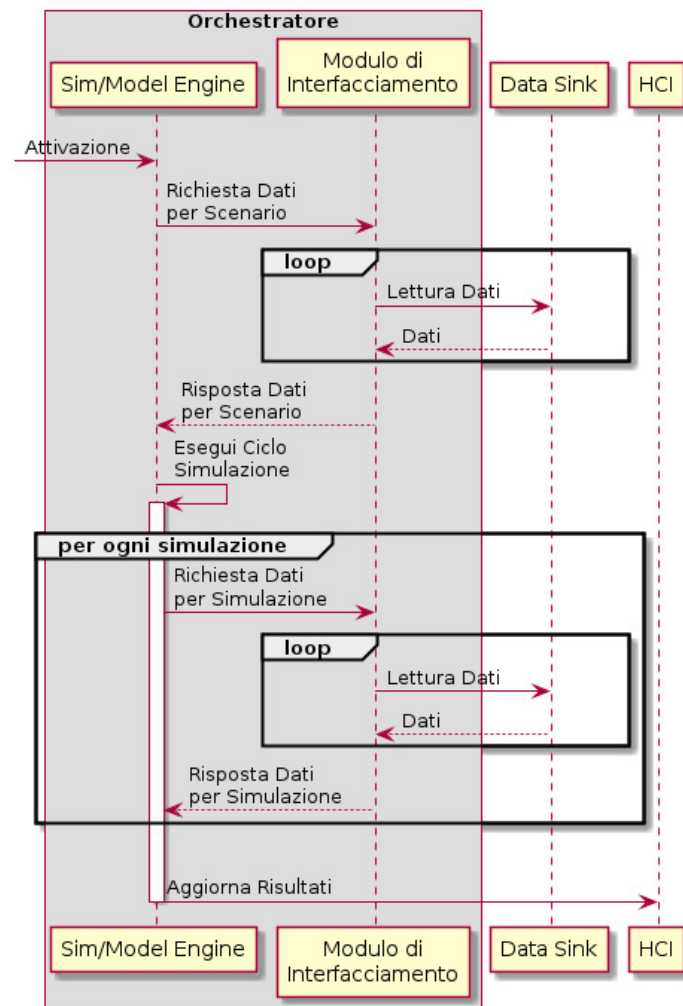


Figura 6 : Sequence Diagram del Simulation/Model Engine

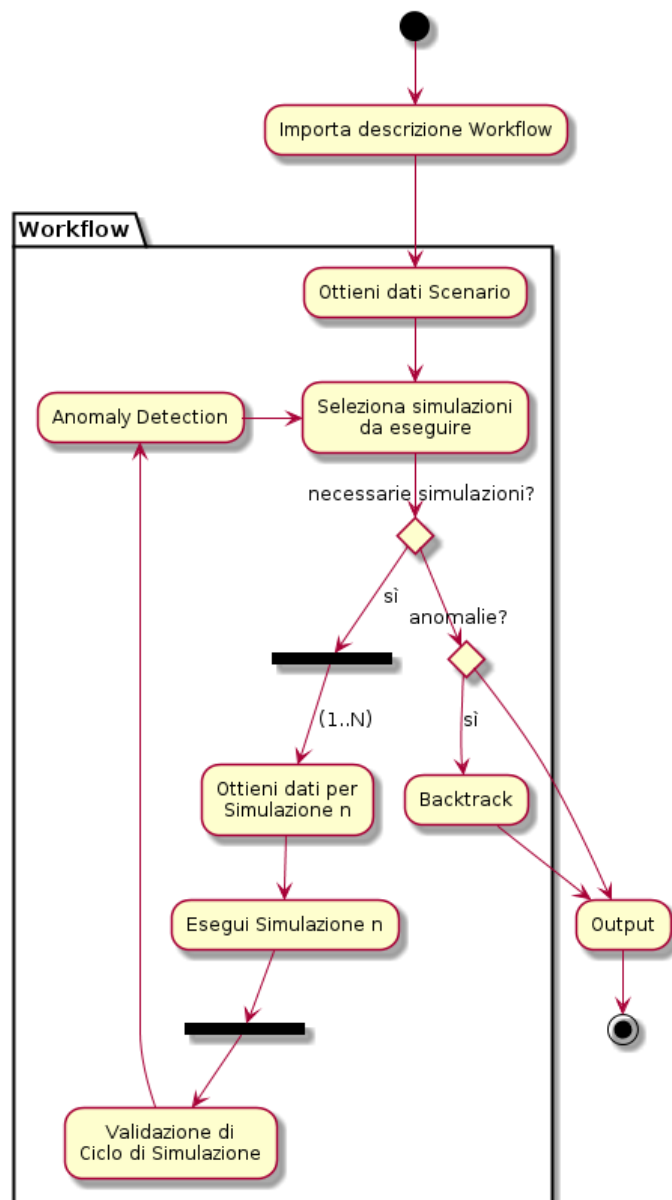
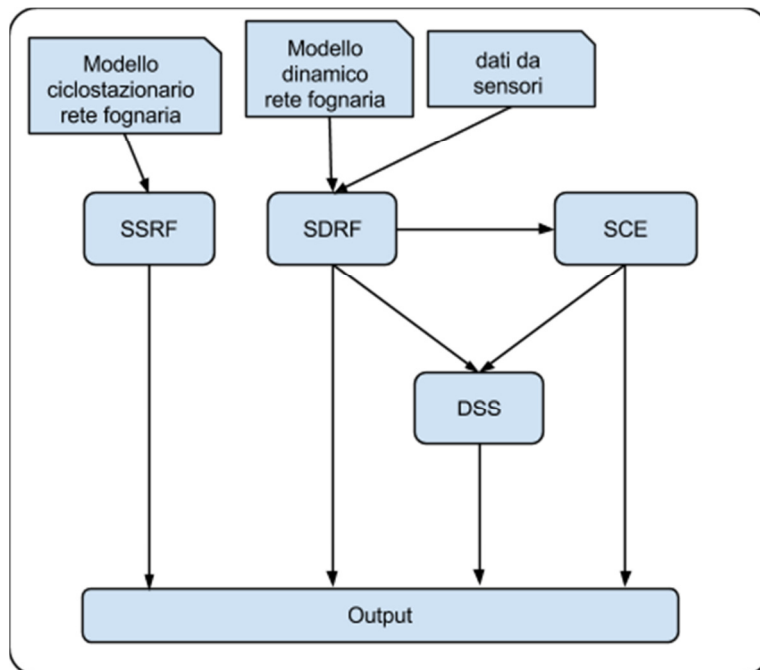


Figura 7: Activity Diagram del Simulation/Model Engine

5.31. Workflow Logic

La funzionalità caratterizzante il Simulation/Model Engine è l'integrazione di simulatori eterogenei tenendo conto di dipendenze funzionali complesse, dinamiche e specificabili con minima necessità di sviluppo software. A tal fine si è scelto di adottare una logica basata sui workflow [leaf] e [yawl], in cui un'elaborazione complessa è descritta in termini di grafo di dipendenze di task ed interfacce di input/output tra task. Tale logica sarà implementata dal modulo Workflow Logic (WL), descritto graficamente nel Component Diagram in Figura 5.

Lo specifico workflow da eseguire, comprendente la descrizione delle dipendenze funzionali e la sequenza di attivazione dei differenti moduli, è descritto in un file in formato XML (parte dell'ontologia) che viene interpretato all'inizio dell'esecuzione del modulo WL (Activity Diagram in Figura 7). Un esempio di workflow è mostrato nel diagramma in Figura 8, in cui le frecce rappresentano il passaggio di dati e quindi una



dipendenza funzionale del tipo produttore-consumatore.

Figura 8 Esempio di workflow gestito dal modulo WL

6 Decision Support System

Il Decision Support System (DSS) sarà implementato come un insieme di modelli Matlab/Octave. Esso riceve come dati di ingresso le letture da sensori e lo stato simulato di alcune componenti della Rete Fognaria e genera una lista di allarmi al superamento di valori di soglia. Il modello è caratterizzato dalla valutazione complessiva dello stato di differenti componenti allo stesso tempo, avendo visibilità totale sui valori rilevati e sui risultati di simulazione.

7 Simulatore Statico di Rete Fognaria

Il Simulatore Statico di Rete Fognaria (SSRF) è una realizzazione del componente astratto *:Simulatore* (si veda il Component Diagram in Figura 5).

Il SSRF verrà implementato utilizzando il software Open Source SWMM5 [swmm]. Il modello che realizza il SSRF simula una rete idrica come un processo ciclostazionario. In tale modello, una volta definiti i parametri strutturali e la presenza o assenza di piogge, i parametri che determinano lo stato del sistema fognario dipendono unicamente dallo stato iniziale, dall'orario di inizio della simulazione e la durata del tempo simulato. In questa particolare simulazione una volta definite queste variabili, l'output resta specificato dalla localizzazione dei punti di misura virtuale rilevanti e dall'ora di misurazione potendosi il processo ripetersi, permanendo lo stato di pioggia/non pioggia *ad libitum*.

8 Simulatore Dinamico di Rete Fognaria

Il Simulatore Dinamico di Rete Fognaria (SDRF) è una realizzazione del componente astratto *:Simulatore* (si veda il Component Diagram in Figura 5).

Il SDRF verrà implementato utilizzando il software Open Source SWMM5 [swmm]. Il modello che realizza il SDRF simula lo stato di un sottografo della rete fognaria, a valle di piattaforme multisensoriali reali. Dati reali provenienti dai sensori costituiscono i parametri d'ingresso della simulazione. Un sistema integrato con SWMM5 caratterizzante i fenomeni di trasporto e le cinetiche di reazione lungo la linea di collettazione permetterà di prevedere il valore di variabili qualitative significative in punti di misura "virtuali" in particolari istanti di tempo successivi alla rilevazione. Questi istanti rappresentano dunque l'orizzonte temporale della simulazione e sono rilevanti ai fini della previsione di eventuali superamenti di soglie critiche in punti prestabiliti come ad esempio l'ingresso agli impianti di depurazione.

9 Simulatore di Consumo Energetico

Il Simulatore di Consumo Energetico (SCE) è una realizzazione del componente astratto *:Simulatore* (si veda il Component Diagram in Figura 5).

Il modello che realizza il SCE descrive il consumo energetico di dispositivi alimentati elettricamente associati alla rete fognaria (es. pompa principale ed ausiliaria di impianti di sollevamento). I dati reali provenienti dai sensori o da altre simulazioni costituiscono i parametri d'ingresso del SCE, insieme alla descrizione dello stato di funzionamento dei dispositivi simulati; in uscita vengono prodotti i consumi previsti per ciascun dispositivo. Il SCE sarà implementato come modello per applicazione di calcolo scientifico Matlab/Octave.

Riferimenti

- [awk] The AWK Programming Language <http://plan9.bell-labs.com/cm/cs/awkbook/>
- [bash] The GNU BASH command interpreter <https://www.gnu.org/software/bash/>
- [curl] cURL, <http://curl.haxx.se>
- [galib] GALib - C++ Genetic Algorithms Library <http://sourceforge.net/projects/galib/>
- [json] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.
- [leaf] Francesco Napolitano, Renato Mariani-Costantini, e Roberto Tagliaferri. *Bioinformatic Pipelines in Python with Leaf*. BMC Bioinformatics 14, n. 1 (2013): 201. doi:10.1186/1471-2105-14-201.
- [nosql] Professional NoSQL, Shashank Tiwari. Wrox, 1 edizione (2011)
- [pyf] Workflow-based programming in Python: <http://pyfproject.org>
- [pymongo] libreria PyMongo : <http://api.mongodb.org/python/current/index.html>
- [pyleaf] python Leaf library for design and management of data flows :
<http://www.francesconapolitano.it/leaf>
- [python] The Python Software Foundation <https://www.python.org>
- [requests] libreria Python per servizi RESTful: <http://docs.python-requests.org>
- [swmm] Storm Water Management Model <http://www2.epa.gov/water-research/storm-water-management-model-swmm>
- [swmmtoolbox] SWMM toolbox utility: <https://pypi.python.org/pypi/swmmtoolbox>
- [swmm-usermanual] SWMM5 User's Manual: <http://nepis.epa.gov/Adobe/PDF/P100ERK4.pdf>
- [yawl] Van Der Aalst, Wil MP, and Arthur HM Ter Hofstede. "YAWL: yet another workflow language." Information systems 30.4 (2005): 245-275.
- [52northAPI] 52north Sensor Event Service interface,
<https://wiki.52north.org/bin/view/SensorWeb/SensorEventServiceInterface>
- [52northInsertObs]
<https://wiki.52north.org/pub/SensorWeb/SosTransactionalProfile/InsertObservation.xml>