

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

UNIVERSITÀ DEGLI STUDI DI ROMA
TOR VERGATA

Corso di Laurea in Informatica

**Linguaggi e strumenti per la specifica di
processi di business basati su IoT**

Relatore:

Andrea D'Ambrogio

Candidato:

Donato Francesco Pio Stanco

ANNO ACCADEMICO 2020/2021

Indice

1	Introduzione al BPM	3
1.1	Concetto di metamodello	4
1.2	Creazione di un processo di Business	5
2	Modellazione dei BP	7
2.1	Organizzazione dei dati in BP	7
2.2	Reti di Petri	8
2.3	Processi guidati dai dati	9
2.3.1	Case models	9
3	BPMN	11
3.1	Definizione dello standard	11
3.2	Analisi elementi di un diagramma BPMN	12
4	IoT e ontologie	14
4.1	Concetto di ontologia	14
4.2	Ontologie nel mondo IoT	14
4.2.1	Ontologia IoT-Lite	15
4.2.2	Ontologia ORDM	16
4.3	OGC SWE	17
4.3.1	SWE Common Data Model	17
4.3.2	SWE Service Model	18
5	Metamodello SWE_BPMN	20
5.1	Perché estendere il BPMN?	20
5.2	Estensione SWE_BPMN	20
5.3	TextAnnotation e definizione sintassi EBNF	23
5.4	Dispositivi IoT nei processi di business	26

6	Applicazioni dell'estensione SWE_BPMN	28
6.1	Supply chain	28
6.2	Flower distribution	31
6.3	Vantaggi dell'estensione	33
7	Processi di business che non usano IoT	34
7.1	Remote Healthcare monitor system	34
7.2	SWE_BPMN nel processo sanitario	37
7.3	Considerazioni finali	40
	Riferimenti	41

Capitolo 1

Introduzione al BPM

Il Business Process Management (BPM) [9] ha ricevuto recentemente una notevole attenzione sia dall'amministrazione aziendale che dalle comunità informatiche. Si basa sull'osservazione che ogni prodotto che un'azienda fornisce al mercato è il risultato di una serie di attività svolte. I Business Process o processi di business sono lo strumento chiave per organizzare queste attività e migliorare la comprensione delle loro interrelazioni. Ma cos'è un Business Process? Un Business Process consiste in un insieme di attività che vengono svolte in coordinamento in un ambiente organizzativo e tecnico, tali attività realizzano congiuntamente un obiettivo aziendale. Ogni processo di business viene attuato da una singola organizzazione, ma può interagire con i processi eseguiti anche da altre organizzazioni. Pertanto, il BPM include concetti, metodi e tecniche per supportare la progettazione, l'amministrazione, la configurazione, l'attuazione e l'analisi dei processi di business.

Tradizionalmente, questi processi sono attuati manualmente, guidati dalla conoscenza del personale dell'azienda e assistiti dalle norme e procedure organizzative che vengono installate. Le aziende possono ottenere ulteriori vantaggi se utilizzano sistemi software per il coordinamento delle attività coinvolte nei processi di business, si parla quindi di Business Process Management System ovvero un sistema software generico guidato da rappresentazioni esplicite dei processi per coordinare l'attuazione di quest'ultimi.

L'ordinamento dei processi può essere utilizzato come un modello che organizza il lavoro, si parla di Business Process Model, un insieme di modelli di attività e vincoli di esecuzione tra loro. Ne consegue la definizione di Business Process Instance, che rappresenta un caso concreto nell'attività operativa di un'azienda, costituita da istanze di attività. Ogni Business Process Model funge da modello per un insieme di Business Process Instance ed ogni modello di attività funge come un modello per l'insieme di istanze dell'attività mettendole perciò in relazione.

Non si deve fare confusione, il termine Business Process viene utilizzato per fare riferimento a Business Process Model o Business Process Instance. Analogamente, l'attività viene utilizzato per riferirsi a modelli di attività o istanze di attività.

1.1 Concetto di metamodello

In base ai livelli di astrazione identificati dal Object Management Group (OMG), il livello del metamodello, il livello del modello e il livello dell'istanza giocano un ruolo importante nella progettazione e analisi di sistemi complessi in generale e sistemi software in particolare. Si parte da un'analisi bottom-up, il livello dell'istanza rappresenta le entità coinvolte nel processo di business, quindi le attività eseguite, i valori concreti dei dati, le risorse e le persone sono presenti in questo livello.

Mentre per organizzare la complessità che comporta un processo di business, un insieme di attività simili a livello dell'istanza vengono identificate e classificate a livello del modello. Differenti livelli di astrazione e le loro relazioni sono riportati in Figura 1.1.

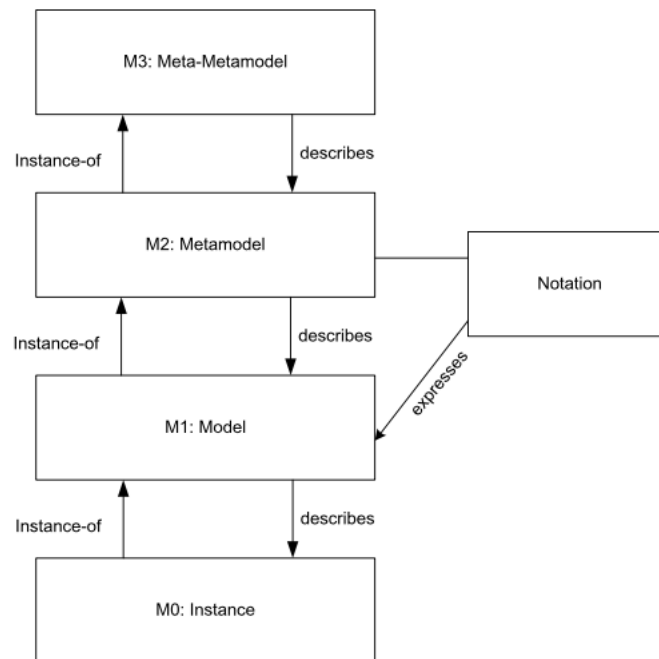


Figura 1.1: Livelli di Astrazione

Si può evidenziare che i modelli sono espressi in metamodelli associati a notazioni, spesso di natura grafica. Ad esempio, il metamodello della rete di Petri definisce le "reti di Petri" in modo che siano costituite da luoghi e transizioni che formano un grafo bipartito diretto. La tradizionale notazione a rete di Petri associa i simboli grafici

agli elementi del metamodello, si ha ad esempio che i luoghi sono rappresentati da cerchi, le transizioni da rettangoli e la struttura del grafo da archi diretti, ma queste notazioni potrebbero cambiare.

1.2 Creazione di un processo di Business

La catena del valore di Porter utilizzata per esaminare il vantaggio competitivo di un'impresa fornisce un'organizzazione di alto livello delle funzioni che quest'ultima svolge. Per avere una visione più dettagliata delle funzioni di business di alto livello, è necessario suddividerle in funzioni di granularità minore e in un'ultima analisi farle diventare processi di business attivi, per far ciò si utilizza la scomposizione funzionale.

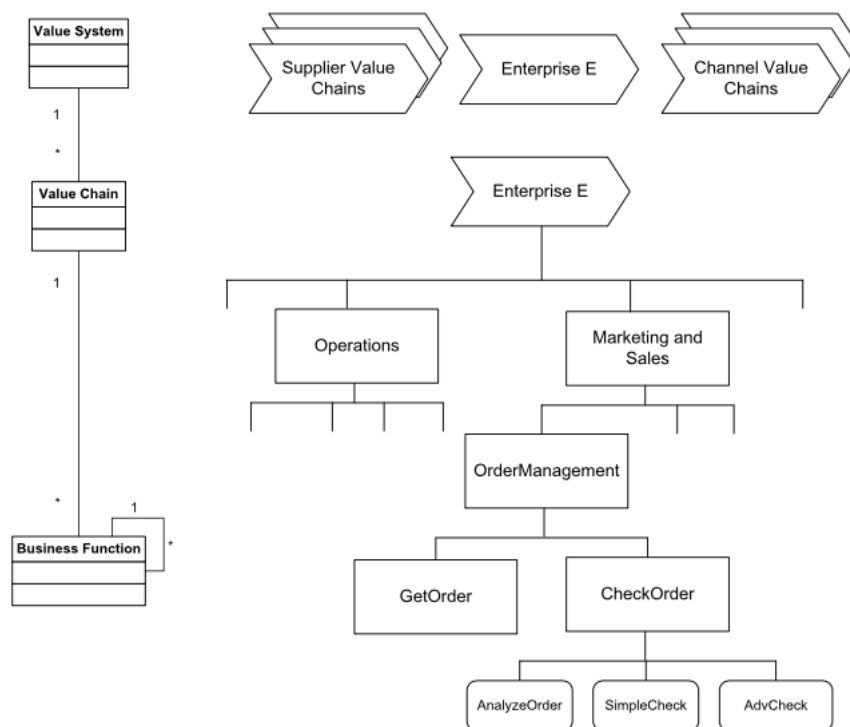


Figura 1.2: Esempio di scomposizione funzionale

Una scomposizione funzionale parziale di una catena del valore è mostrata nella Figura 1.2, dove un sistema di valori rappresenta il più alto livello di aggregazione. Ogni sistema di valori è costituito da una serie di catene di valore, caratterizzate dal diagramma delle classi sul lato sinistro.

Come mostrato nella Figura 1.2, ci sono diversi simboli per le funzioni aziendali e per le funzioni di massima granularità: le funzioni aziendali sono rappresentate da rettangoli, mentre le funzioni di massima granularità sono rappresentate da rettangoli

con angoli arrotondati. Le funzioni a livello foglia, elementi che si trovano sull'ultimo livello della scomposizione funzionale, sono anche chiamate attività. Tradizionalmente, la scomposizione funzionale veniva utilizzata per descrivere le imprese in base alle funzioni che svolgono. A questo punto i processi di business operativi mettono in relazione le attività, introducendo dei vincoli di esecuzione tra loro. In linea di principio, le funzioni relative ai processi di business possono essere applicate in base alla granularità delle funzioni aziendali.

Capitolo 2

Modellazione dei BP

I processi di business operano sui dati. La modellazione dei dati è il cuore del design dei database ed utilizza l'approccio dell'Entity Relationship per classificare ed organizzare un determinato dominio dell'applicazione. Ovviamente questo processo di modelling può essere utilizzato per rappresentare le strutture dati in BPM. Sulla base di queste strutture dati, le dipendenze dei dati tra le attività nei processi di business possono essere acquisite con precisione.

2.1 Organizzazione dei dati in BP

Per organizzare i problemi relativi ai dati nella gestione dei processi di business, sono stati introdotti dei workflow data patterns (pattern di dati del flusso di lavoro). Questi pattern formulano caratteristiche su come gestire i dati nei processi di business. Sono organizzati in base alla visibilità, all'interazione, al trasferimento e all'instradamento dei dati. La visibilità dei dati è molto simile all'ambito dei linguaggi di programmazione perché caratterizzano l'area in cui un certo oggetto è disponibile per l'accesso. I workflow data patterns più importanti sono:

- Task data: l'oggetto dati è locale per una particolare attività; non è visibile ad altre attività dello stesso processo o ad altri processi.
- Block data: l'oggetto dati è visibile a tutte le attività di un determinato sotto processo.
- Workflow data: l'oggetto dati è visibile a tutte le attività di un determinato processo aziendale, ma l'accesso è limitato dalla gestione del processo aziendale.

- Environment data: l'oggetto dati fa parte del processo aziendale eseguito nell'ambiente operativo; è possibile accedervi dalle attività del processo durante la sua attuazione.

Questi pattern di interazione dei dati descrivono come quest'ultimi devono essere passati tra le attività e i processi. I dati possono essere comunicati tra attività dello stesso processo di business, tra attività e sotto processi dello stesso processo di business e anche tra attività di processi di business diversi. I dati possono anche essere comunicati tra il processo di business e il BPM. Pertanto, i workflow data patterns sono un mezzo appropriato per organizzare gli aspetti dei processi di business relativi alla gestione dei dati.

2.2 Reti di Petri

Le reti di Petri sono una delle tecniche più conosciute per rappresentare i processi di business in modo formale e astratto. Queste reti possono essere utilizzate per modellare sistemi dinamici con una struttura statica. La struttura statica è rappresentata da una rete di Petri, e il comportamento dinamico è catturato dal token (gettone) della rete. La dinamica del sistema rappresentata da una rete di Petri è modellata da token che risiedono sui luoghi. Mentre la struttura delle reti di Petri è fissa, i token possono cambiare la loro posizione in base alle regole di tiro (fire). L'attuale distribuzione dei token tra i luoghi determina lo stato della rete di Petri e, quindi, del sistema da essa modellato, il movimento di tali token è chiamato token play. Poiché le transizioni possono modificare lo stato di una rete di Petri, esse sono considerate componenti attive, che tipicamente rappresentano eventi, operazioni, trasformazioni o trasporti.

Un luogo è una componente passiva, che rappresenta un mezzo, un cuscinetto, uno stato o una condizione. I token vengono utilizzati per rappresentare oggetti fisici o oggetti informativi. Nel contesto dei processi di business, le transizioni rappresentano attività e i luoghi contenenti token rappresentano gli stati delle istanze del processo. Lo stato della rete di Petri è caratterizzato dalla distribuzione dei token sui luoghi della rete. Un esempio di rete di Petri è riportato in Figura 2.1.

Formalmente una rete di Petri è una tripla (P, T, F) con:

- un insieme finito P di posti,
- un insieme finito T di transizioni tale che $T \cap P = \emptyset$,
- una relazione di flusso $F \subseteq (P \times T) \cup (T \times P)$.

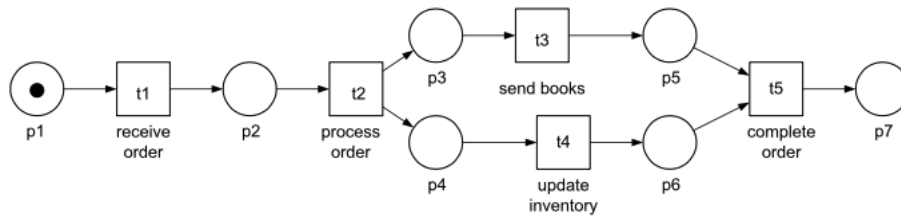


Figura 2.1: Rete di Petri

2.3 Processi guidati dai dati

La gestione dei casi mira a bilanciare l'orientamento al processo con l'orientamento ai dati per controllare l'esecuzione dei processi di business, tale aspetto è derivato dalla re-ingegnerizzazione di quest'ultimi.

Con l'avvento dell'informatica il ruolo del lavoratore è cambiato, poiché il knowledge worker ora è al centro ed è responsabile del proprio lavoro essendo altamente qualificato, può svolgere un'ampia gamma di attività richieste per soddisfare gli obiettivi aziendali dell'azienda. L'idea alla base della gestione dei casi può essere descritta nel modo seguente, si considerino le attività A e B di un processo di business ordinate dal control flow (flusso di controllo) $A \rightarrow B$. Si evince che B può essere abilitata (e quindi può essere avviata) solo dopo che A è terminata. Sebbene in molti scenari di processi di business questo approccio tradizionale relativo al workflow sia adeguato, in domini ad alta intensità di conoscenza, dove un ruolo attivo del knowledge worker guida il processo, sono necessari approcci più flessibili. Ad esempio, l'attività A non crea i dati al termine, ma durante l'esecuzione. Supponiamo inoltre che B possa iniziare a lavorare una volta che i valori dei dati creati da A sono disponibili. Quindi, B può iniziare a lavorare su questi dati, mentre A crea i valori dei dati rimanenti. In questo caso, il vincolo del control flow tra A e B limita un ordine di esecuzione utile, in cui B inizia a lavorare prima che A completi.

2.3.1 Case models

Per gestire i vari casi di un processo di business è necessaria una definizione granulare delle dipendenze tra i dati. Questa definizione si basa sulla disponibilità dei dati e sui vincoli relativi ai dati per le attività. In tal modo, il modello di processo contiene informazioni su quali dati sono obbligatori per l'avvio dell'attività successiva e quali sono facoltativi.

Questa conoscenza specifica del dominio deve essere fornita durante la progettazione delle applicazioni di gestione dei casi. Poiché il caso è al centro della gestione dei casi, le rappresentazioni astratte dei casi sono note come case models (modelli di caso). Un modello di caso contiene attività, oggetti dati e relazioni tra loro, nonché ruoli diversi e altre informazioni organizzative e operative. Un esempio di gestione dei casi e delle dipendenze tra i dati è riportato in Figura 2.2.

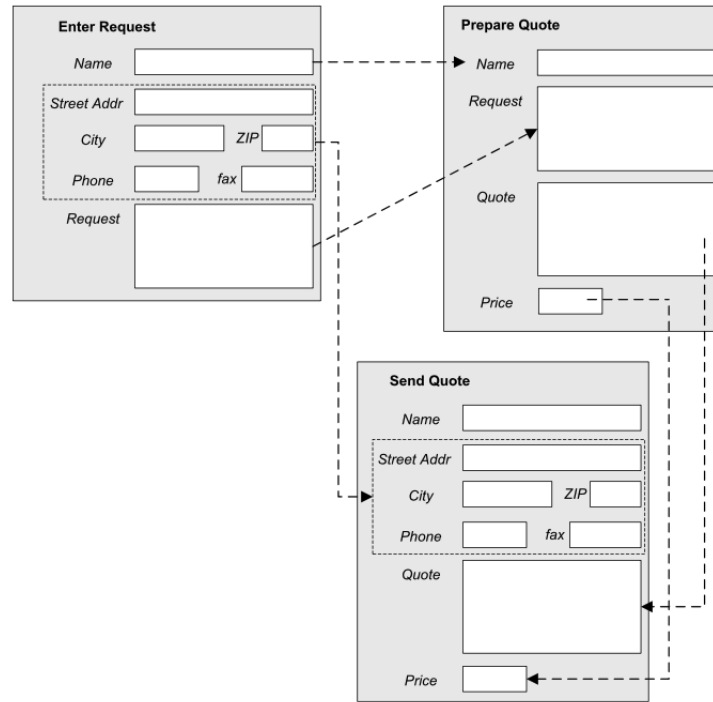


Figura 2.2: Esempio gestione dei casi e dipendenza dati

Pertanto, rappresentando nei case models le dipendenze dei dati a grana fine associate alle attività condotte dai knowledge worker, è possibile consentire ulteriori esecuzioni valide senza violare la coerenza complessiva del processo di business.

Capitolo 3

BPMN

Il Business Process Model and Notation è diventato lo standard de facto per la modellazione dei processi di business (BP). Tale linguaggio di modellazione è definito dal Object Management Group (OMG) e specificato come standard ISO ed è ampiamente usato per la sua espressività, semplicità e ricchezza semantica.

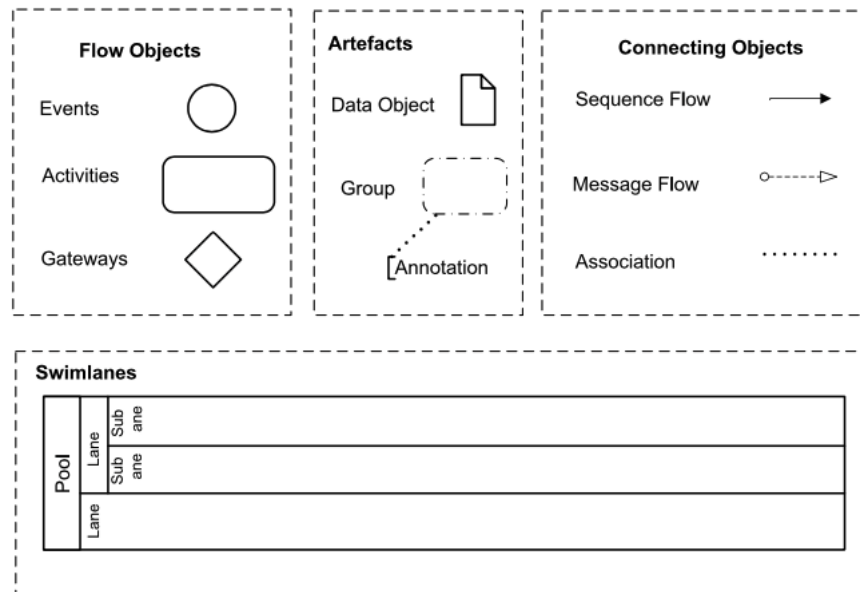
3.1 Definizione dello standard

Lo standard BPMN definisce una notazione e un metamodello che organizza i concetti utilizzati nel linguaggio di modellazione.

Gli elementi di modellazione di base del BPMN consentono di esprimere strutture semplici nei processi di business. La notazione grafica in questi processi è completata da una serie di attributi che possono essere associati al diagramma di processo completo e ad elementi particolari.

Il BPMN assomiglia più a un framework piuttosto che a un linguaggio concreto, perché alcuni aspetti, ad esempio le espressioni, non sono coperti dallo standard e sono lasciati al progettista del processo. Gli elementi notazionali nei diagrammi dei processi di business sono suddivisi in quattro categorie di base, ciascuna delle quali consiste in un insieme di elementi, come riportato in Figura 3.1.

Gli artefatti vengono utilizzati per mostrare informazioni aggiuntive su un processo di business che "non è direttamente rilevante per il flusso di sequenza o il flusso di messaggi del processo", come menzionato dallo standard. Oggetti, gruppi e annotazioni sono artefatti supportati. Ogni artefatto può essere associato a elementi di flusso. Gli artefatti servono solo a scopo informativo, in modo che la semantica di esecuzione di un processo non ne sia influenzata. I data object sono rappresentati semplicemente da un nome; la struttura interna di questi non può essere definita in BPMN. Lo scopo principale di tali artefatti è la documentazione dei dati utilizzati nel processo.

**Figura 3.1: Elementi BPMN**

Le annotazioni di testo documentano aspetti specifici del processo di business in forma testuale ed il testo è graficamente associato all'oggetto nel diagramma del processo. Gli oggetti di gruppo sono artefatti utilizzati per raggruppare gli elementi di un processo. I gruppi non hanno un significato formale; servono solo a scopi di documentazione. Gli oggetti di collegamento connettono oggetti del flusso, swimlanes o artefatti. Il flusso di sequenza viene utilizzato per specificare l'ordine degli oggetti del flusso, mentre il flusso di messaggi descrive il flusso di messaggi tra i business partner rappresentati dai pool. L'associazione è un tipo specifico di oggetto di connessione utilizzato per collegare gli artefatti agli elementi nei diagrammi dei processi di business.

3.2 Analisi elementi di un diagramma BPMN

Un esempio di diagramma BPMN è riportato in Figura 3.2.

Si possono notare alcuni elementi in Figura 3.2, i rettangoli con angoli arrotondati rappresentano le attività che descrivono il particolare lavoro svolto in un'istanza di processo ed esistono quattro tipi di attività BPMN: compiti, sotto-processi (attività che si adattano particolarmente insieme), transizioni (sotto-processo specializzato) e chiamata (sotto-processo globale riutilizzato in vari punti del flusso). I rombi sono i simboli di gateway che separano e ricombinano i flussi, il cerchio indica un evento (in questo caso il cerchio all'inizio è l'evento iniziale ed indica un messaggio, mentre

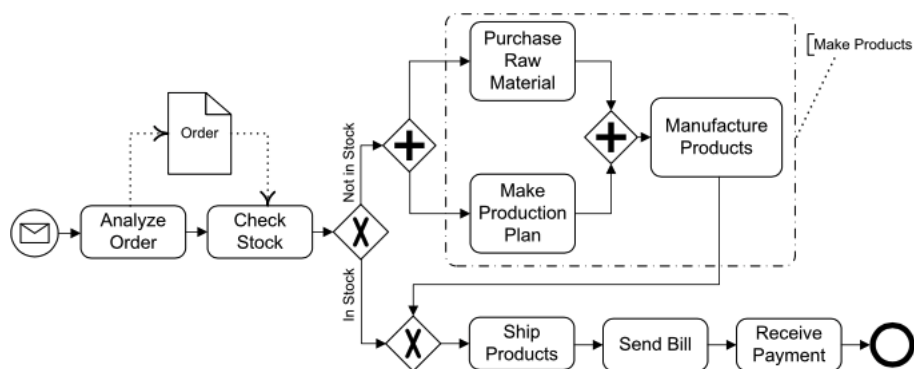


Figura 3.2: Diagramma di un processo di business espresso in BPMN

quello alla fine del diagramma indica l'evento finale) ed infine il simbolo a forma di foglio rappresenta il data object.

Capitolo 4

IoT e ontologie

La prima menzione di Internet of Things (IoT) risale al 1999, quando Kevin Ashton del Massachusetts Institute of Technology (MIT) propose l'idea di aggiungere tag RFID (Radio-Frequency Identification) agli oggetti di uso quotidiano. Inizialmente, la rivoluzione targata IoT è cominciata con l'utilizzo di questi tag, sensori e reti di sensori per il monitoraggio degli oggetti, specialmente nella catena di approvvigionamento.

Lo sviluppo tecnologico degli ultimi dieci anni ha favorito il diffondersi di questi oggetti diventando popolari e accessibili a tutti. Oggi, molti di questi dispositivi consentono l'interazione con il mondo fisico esponendo le loro funzionalità tramite servizi standard e sono considerati come i promotori per la visione di un mondo intelligente. Ovviamente la crescita dell'utilizzo di dispositivi IoT ha aumentato la complessità coinvolta nella loro gestione, a causa di caratteristiche come sensibilità, tempo di risposta e vincoli di risorse (capacità di calcolo, capacità energetica).

4.1 Concetto di ontologia

Un'ontologia è una rappresentazione formale della conoscenza e consiste di affermazioni che definiscono concetti, relazioni e vincoli. È analoga a un diagramma delle classi orientato agli oggetti e forma un modello di dominio dell'informazione. Inoltre, consente una comprensione comune condivisa, il riutilizzo della conoscenza del dominio e anche l'analisi di quest'ultima.

4.2 Ontologie nel mondo IoT

Nel mondo IoT esistono una grande varietà di ontologie, tra cui si hanno:

- Ontologia IoT-Lite;

- Ontologia ORDM per le risorse IoT;
- OGC SWE (Sensor Web Enablement).

4.2.1 Ontologia IoT-Lite

L'ontologia IoT-Lite [1] è stata progettata per essere scalabile ed ha bisogno del concetto di sensore, della qualità che misura e di un endpoint. Lo schema ontologico è rappresentato in Figura 4.1.

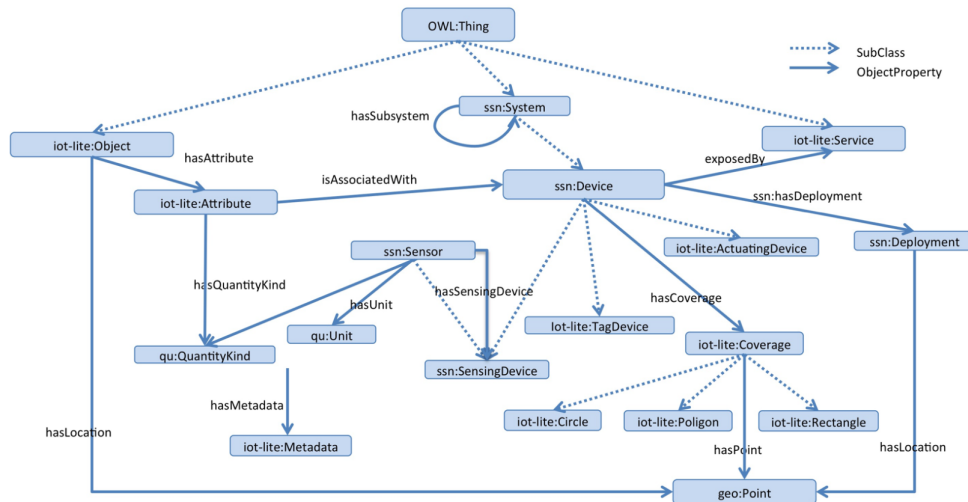


Figura 4.1: Schema ontologico IoT-Lite

Questo tipo di ontologia è interoperabile ovvero collegata ad altre ontologie come quella SWEET e SSN, evitando quelle poco utilizzate. Pertanto, per realizzarla è stata creata un'istanza di SSN, così da poter creare un'ontologia leggera con i concetti principali, utilizzando i tre elementi cardini nella classificazione delle entità IoT:

- Entità o oggetti;
- Risorse o dispositivi;
- Servizi.

Quindi per creare l'ontologia sono stati sfruttati questi tre concetti avendo le seguenti relazioni: un oggetto (o entità) *iot-lite:Object* ha un attributo *iot-lite:Attribute* che è associato a un dispositivo (o risorsa) *iot-lite:Device*, che è esposto da un servizio *iot-lite:Service*. In conclusione, questo tipo di ontologia permette di utilizzare

gli aspetti più leggeri del modello semantico SSN basandosi sui tre elementi cardini, tuttavia per avere un modello completo è necessaria l'implementazione di un ulteriore schema semantico aggiuntivo.

4.2.2 Ontologia ORDM

Nel mondo IoT una descrizione unificata delle risorse è il prerequisito per la realizzazione della condivisione delle informazioni e dei dati tra dispositivi. L'ontologia ORDM [8] protegge l'eterogeneità dei dispositivi IoT, attraverso l'astrazione delle entità e metodi di descrizione unificata per ottenere la condivisione di informazioni e dati tra loro. Lo schema ontologico è riportato in Figura 4.2.

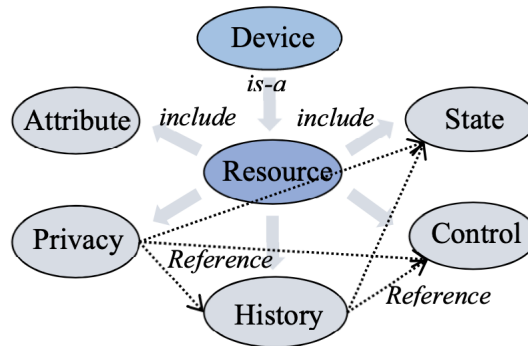


Figura 4.2: Schema ontologico ORDM

Anche in questa ontologia vengono utilizzati i tre elementi fondamentali nella classificazione IoT ovvero: entità, dispositivi e servizi. Il modello ontologico utilizzato è quello di Perez che divide gli elementi di modellazione in cinque linguaggi originali di modellazione di base: classi, relazioni, funzioni, assiomi e istanze.

$$O = C, R, F, A, I$$

Il contenuto della descrizione per i dispositivi IoT può essere suddiviso in diversi aspetti:

- Le informazioni intrinseche del dispositivo;
- Le informazioni sulla produzione e sul feedback;
- Le informazioni di autenticazione e privacy.

Questo tipo di ontologia richiede ogni volta la progettazione delle classi mostrate in Figura 4.2 e ciò può risultare laborioso dal punto di vista progettuale.

4.3 OGC SWE

L'organizzazione internazionale Open Geospatial Consortium (OGC) ha definito due standard per lo scambio di dati per i sensori all'interno del framework OGC SWE:

1. SWE Common Data Model
2. SWE Service Model

4.3.1 SWE Common Data Model

Questi modelli consentono alle applicazioni e/o ai server di strutturare, codificare e trasmettere set di dati dei sensori in modo autodescrittivo e semanticamente abilitato. I pacchetti di questo modello sono riportati in Figura 4.3.

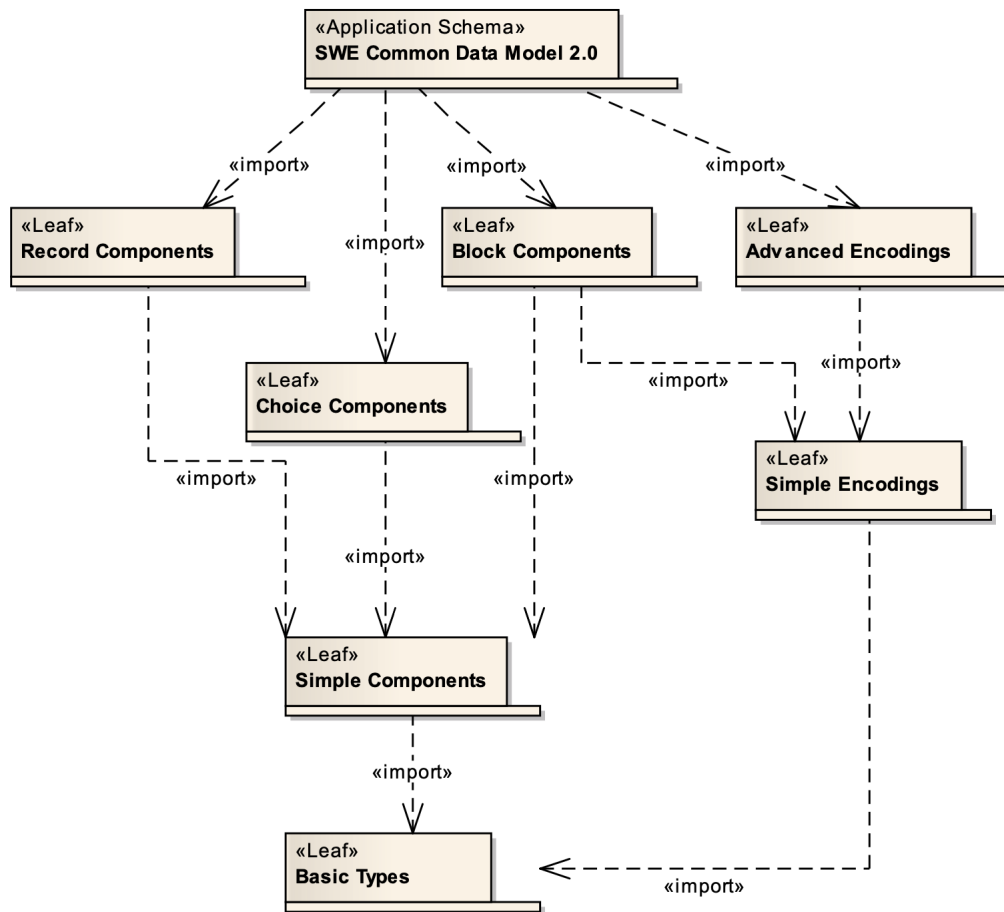


Figura 4.3: Overview pacchetti SWE Common Data Model

Il Common Data Model SWE [5] è stato concepito per essere utilizzato per descrivere dati statici (file) e set di dati generati dinamicamente, sottoinsiemi di dati, input e

output di processi e servizi Web e dati in streaming in tempo reale. Più precisamente copre i seguenti aspetti della descrizione dei set di dati:

- Rappresentazione;
- Natura dei dati e della semantica;
- Qualità;
- Struttura;
- Codifica.

Vi sono una serie di classi che permettono di gestire i dati sia di tipo semplice come boolean, text e category, classi per la gestione di array, matrici e stream di dati e classi per la codifica come TextEncoding, XMLEncoding e BinaryEncoding.

4.3.2 SWE Service Model

Questo standard specifica i dati e le interfacce comuni ai servizi Web di sensori [4], attualmente definisce otto pacchetti con tipi di dati per uso comune tra i servizi SWE. Cinque di questi pacchetti definiscono la richiesta dell'operazione e i tipi di risposta. I pacchetti presenti nel modello sono:

- Contents;
- Notification;
- Common;
- Common codes;
- DescribeSensor;
- UpdateSensorDescription;
- InsertSensor;
- DeleteSensor.

La dipendenza tra i vari pacchetti è mostrata in Figura 4.4.

Questo tipo di modello permette di inserire nuovi sensori utilizzando il pacchetto InsertSensor e i vari parametri vengono passati dal client e sono giudicati sufficienti affinché il servizio includa il nuovo sensore.

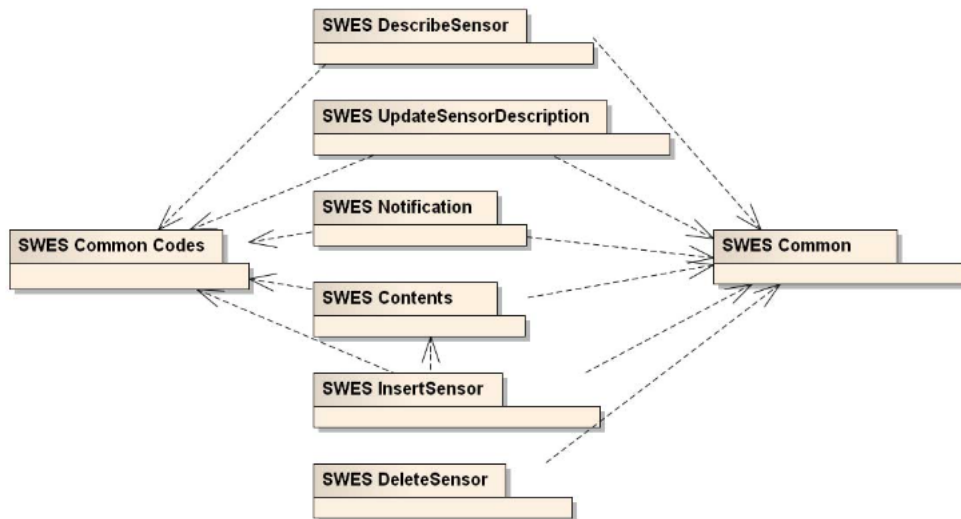


Figura 4.4: Overview delle dipendenze dei pacchetti nel SWE Service Model

È presente anche un pacchetto DeleteSensor ma questo tipo di operazione è astratta anche se i dati di richiesta e risposta non sono astratti, infatti l'operazione può essere utilizzata solo in combinazione con un dato servizio SWE se la specifica di quel tipo di servizio definisce la semantica dell'operazione di cancellazione del sensore.

Quindi in sostanza tutte le classi del Common Data Model vengono implementate nel Service Model in modo che entrambi i modelli comunichino tra loro, offrendo quindi una panoramica completa della semantica utilizzata, in modo da permettere lo scambio dei dati tra sensori all'interno del framework OGC SWE.

Capitolo 5

Metamodello SWE_BPMN

Il BPMN diventa limitato nel supportare domini specifici o proprietà non funzionali perché fornisce elementi generici e si concentra solo sui requisiti funzionali dei BP. Per superare questa lacuna e ampliare i casi d'uso BPMN, OMG ha introdotto un meccanismo di estensione che consente agli utenti di integrare nuovi elementi e fornire estensioni BPMN valide (OMG, 2013) [10].

5.1 Perché estendere il BPMN?

BPMN 2.0.2 fornisce un meccanismo di estensione che consente di rappresentare concetti aggiuntivi e di collegarli ai suoi elementi originali per rappresentare le caratteristiche di un particolare dominio (ad esempio, assistenza sanitaria, gestione della qualità, sicurezza, ecc.).

L'estensione del BPMN e il riutilizzo del suo kernel consentono di sfruttare i suoi vantaggi (ad es. standardizzazione, supporto di strumenti) ed evitare lo sviluppo costoso di un linguaggio di modellazione specifico del dominio (DSML) da zero [3]. Il punto debole nel meccanismo di estensione BPMN è la mancanza di guide metodologiche per lo sviluppo di tali estensioni. Per colmare questa lacuna, è stata proposta una procedura basata su Model-Driven Architecture (MDA) per lo sviluppo metodico di estensioni BPMN valide.

5.2 Estensione SWE_BPMN

L'autore della tesi ha usato l'ontologia definita dall'OGC ovvero quella OGC SWE per estendere il metamodello BPMN [2].

Il metamodello dell'estensione proposta chiamato SWE_BPMN è stato progettato sfruttando standard, principi e metodi raccomandati da MDA, ovvero tecniche di meta-

modellazione che consentono sia la modellazione astratta formale dei modelli che le trasformazioni tra quest'ultimi.

Per poter mettere in atto l'estensione è necessario creare una relazione fra il meta-modello BPMN e quello dell'estensione, tale relazione è riportata in Figura 5.1.

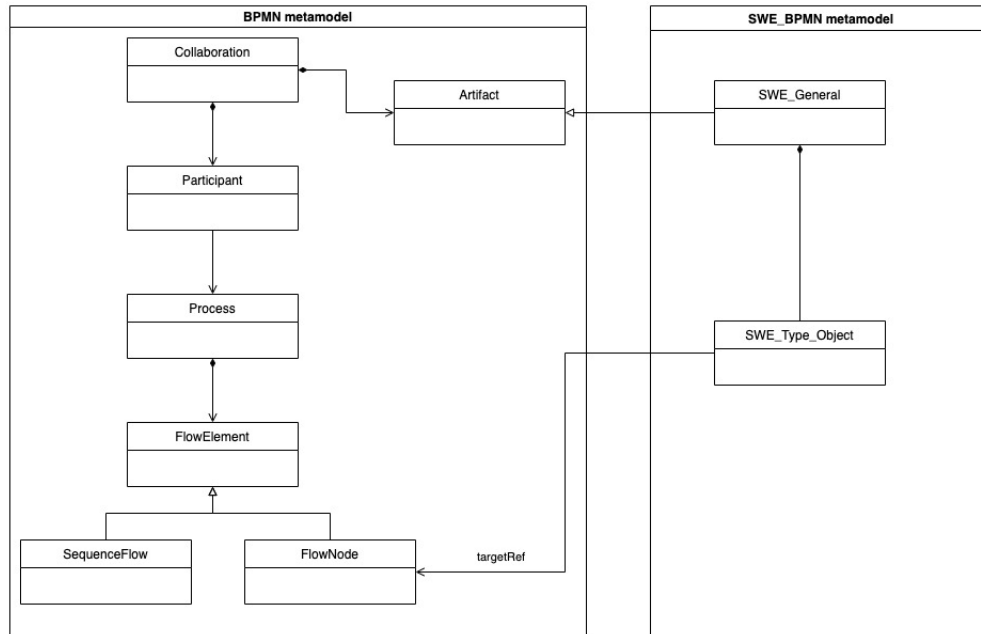


Figura 5.1: Relazione tra metamodello BPMN e metamodello SWE_BPMN

Dalla Figura 5.1 si nota che la metaclass SWE_Type_Object è quella che mette in atto l'estensione, poiché permette di associare le proprietà non funzionali, definite nelle metaclassi dell'estensione, ad elementi BPMN standard di tipo FlowNode. Come detto, BPMN presenta dei limiti sia per quanto riguarda domini che proprietà non funzionali, pertanto l'estensione SWE_BPMN si occupa di evidenziare la proprietà non funzionale rispetto al concetto di data object.

Nella modellazione BPMN i data object sono rappresentati semplicemente da un nome e la loro struttura interna non è definita, per questo come artefatti producono solo una documentazione dei dati utilizzati nel processo di business.

Quindi, l'estensione SWE_BPMN, riportata in Figura 5.2, permette in fase di progettazione di definire i dati coinvolti nelle attività tra i vari sensori nel mondo IoT.

Il metamodello SWE_BPMN può essere prodotto automaticamente partendo dal modello standard BPMN e applicando la trasformazione model-to-model. Gli elementi di base (ovvero gli elementi utilizzati per definire il flusso dei compiti e delle attività) sono parte del metamodello BPMN.

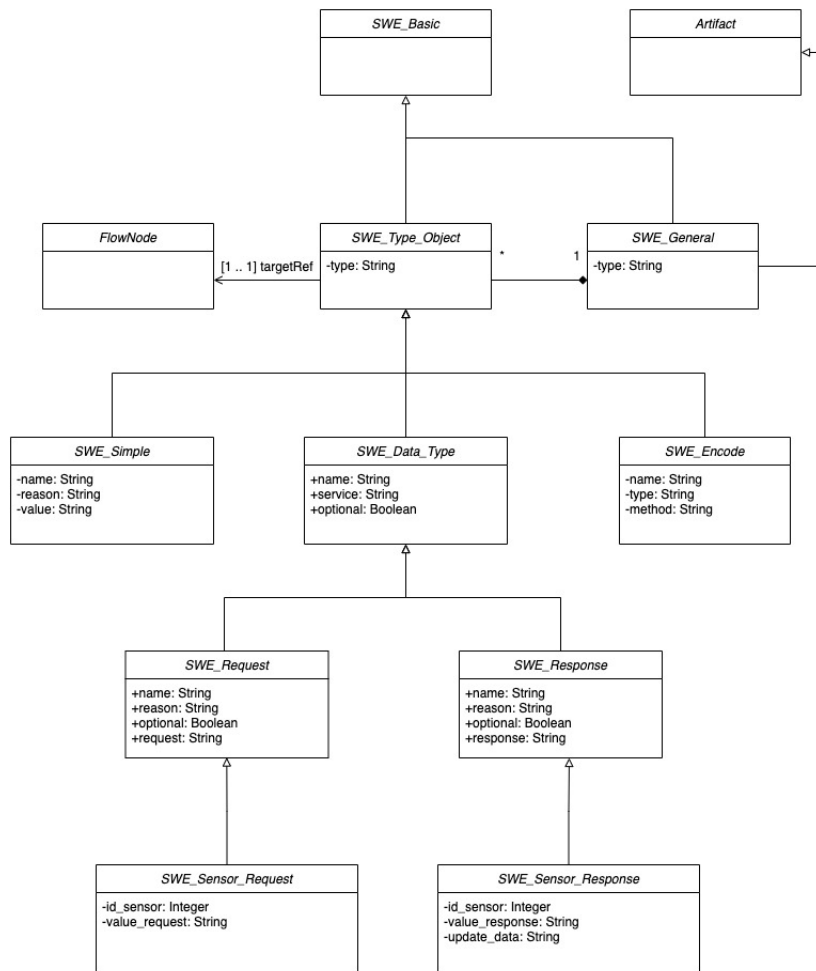


Figura 5.2: Metamodello SWE_BPMN ottenuto applicando la trasformazione model to model

Una breve descrizione delle metaclassi introdotte nel metamodello SWE_BPMN è la seguente:

- **SWE_General:** è una metaclassa che specifica un oggetto SWE e permette di utilizzare gli elementi BPMN standard. Lo scopo di questa metaclassa è di descrivere la generica classe SWE_Basic, che rappresenta l'entità che svolge l'attività.
- **SWE_Type_Object:** è una metaclassa che rappresenta un singolo oggetto SWE.
- **SWE_Simple:** questa metaclassa rappresenta gli oggetti “semplici”, come ad esempio la qualità o l'approssimazione numerica.
- **SWE_Data_Type:** tale metaclassa specifica i tipi di dati che possiamo avere nell'attività che si sta svolgendo.

- SWE.Encode: rappresenta gli oggetti di encode usati per codificare i dati dei sensori.
- SWE.Request: metaclassa che rappresenta la “richiesta” di un’attività fornendo ad esempio il servizio o la versione della descrizione di un sensore (usando la generalizzazione della metaclassa SWE_Sensor_Request).
- SWE.Response: metaclassa che rappresenta la “risposta” di un’attività (usando la generalizzazione della metaclassa SWE_Sensor_Response).

5.3 TextAnnotation e definizione sintassi EBNF

Una TextAnnotation che include una specifica SWE_BPMN è chiamata SWEAnnotation ed ognuna di queste contiene una o più definizioni con la seguente sintassi Extended Backus-Naur Form (EBNF). La grammatica utilizzata per rappresentare un elemento SWE è riportata in Figura 5.3.

```

<SWEParamList> ::= <String> '=' <Value> ( ',' <String> '='
<Value>)*
<Value> ::= <Literal> | <ComplexValue> | <CollectionValue>
<Literal> ::= <String> | <Number>
<ComplexValue> ::= '(' <String> '=' <Literal> ( ',' <String> '='
<Literal>)* ')'
<CollectionValue> ::= '(' <Literal> ( ',' <Literal>)* ')'

```

Figura 5.3: Grammatica in forma EBNF per rappresentare elemento SWE

Ogni elemento SWE è specificato da un’espressione regolare dove gli elementi tra gli apici rappresentano i simboli terminali mentre quelli tra le parentesi angolari sono i simboli non terminali.

Per chiarire l’uso del termine $\langle \text{Value} \rangle$ e di rispettare le relazioni tra le varie meta-classi nel metamodello SWE_BPMN, è possibile seguire le seguenti linee guida:

- gli attributi semplici dei tipi di dati possono essere specificati come $\langle \text{Literal} \rangle$;
- gli attributi che fanno riferimento agli oggetti SWE_BPMN sono specificati come $\langle \text{ComplexValue} \rangle$;

- gli attributi che fanno riferimento ad un insieme di altri elementi sono di tipo $\langle \text{CollectionValue} \rangle$.

A questo punto si deve utilizzare la sintassi EBNF in modo da creare una definizione (SWEDefinition), mostrata in Figura 5.4, per l'estensione SWE_BPMN in modo da avere la seguente sintassi formale:

```
<SWEDefinition> ::= <SWE_General> '{' <SWEParamList> '}'  
  
<SWE_General> ::= '<<SWE_Type_Object>>'  
| '<<SWE_Simple>>'  
| '<<SWE_Data_Type>>'  
| '<<SWE_Encode>>'
```

Figura 5.4: Sintassi formale dell'estensione SWE_BPMN

Gli elementi racchiusi da doppie parentesi angolari da ambo i lati rappresentano le metaclassi dell'estensione e la metaclassa SWE_General va a definire tutte le altre presenti nel metamodello SWE_BPMN.

La rappresentazione degli attributi delle varie metaclassi dell'estensione può essere definita senza perdita di generalità nel seguente modo:

- name, type, reason, value, optional possono essere specificati come un valore di tipo String;
- per SWE_Encode si ha valueofEncode può essere specificato come un valore di tipo $\langle \text{CollectionValue} \rangle$;
- per SWE_Simple si ha che scope può essere specificato come un valore di tipo $\langle \text{CollectionValue} \rangle$;
- per SWE_Data_Type si ha che definition può essere specificato come un valore di tipo $\langle \text{CollectionValue} \rangle$.

Una volta descritte le funzionalità delle varie metaclassi ed aver illustrato i vari attributi che esse contengono, è possibile utilizzare l'estensione SWE_BPMN ed andare a definire le varie metaclassi utilizzando la sintassi EBNF.

La prima metaclassa che si va a definire è quella SWE_Simple che si occupa di gestire lo "stream" dei dati, ed è riportata in Figura 5.5.

Dalla formalizzazione della metaclassa si evince che bisogna definire due attributi, un name e uno scope che viene specificato come un $\langle \text{CollectionValue} \rangle$.

```
<<SWE_Simple>> {  
  name = Nil,  
  scope = (reason = measurement, value = stream_of_data) }
```

Figura 5.5: Definizione della metaclassa SWE_Simple con sintassi EBNF

La seconda metaclassa che si definisce è quella SWE_Data_Type che permette di gestire i dati che vengono ricevuti/intercettati da un sensore durante l'esecuzione del processo di business.

```
<<SWE_Data_Type>> {  
  name = dataDetected,  
  definition = (service = measurement, optional = true) }
```

Figura 5.6: Definizione della metaclassa SWE_Data_Type con sintassi EBNF

Anche in questo caso si definisce un name che rappresenta l'oggetto che si va a creare e un attributo definition specificato come un $\langle \text{CollectionValue} \rangle$.

Infine la terza metaclassa è quella responsabile della codifica dei dati utilizzando uno dei tre metodi messi a disposizione dal Service Model: TextEncoding, BinaryEncoding e XMLEncoding. La definizione della metaclassa SWE_Encode è riportata in Figura 5.7.

```
<<SWE_Encode>> {  
  name = encoding,  
  valueofEncode = (type = string, method = TextEncoding) }
```

Figura 5.7: Definizione della metaclassa SWE_Encode con sintassi EBNF

Come per la dichiarazione delle due metaclassi precedenti anche per questa si definisce un name e un valueofEncode specificato come un $\langle \text{CollectionValue} \rangle$.

5.4 Dispositivi IoT nei processi di business

Quando un processo di business lavora in un ambiente è necessario descriverlo ovvero definire tutti i dispositivi IoT presenti [7], in generale ne abbiamo di due tipi:

- sensori che si occupano di rilevare dei dati e di scambiarli tra di loro all'interno di una rete sensoristica;
- tag RFID che sono dei piccoli chip, in cui sono contenute le informazioni riguardanti l'articolo o il prodotto che identifica, il tutto viene codificato attraverso radiofrequenza (LF, HF o UHF).

Pertanto, bisogna definire formalmente utilizzando la sintassi EBNF il concetto di sensore e per poterlo specificare si devono rappresentare i due oggetti delle metaclassi SWE_Sensor_Request e SWE_Sensor_Response, quindi utilizzando la grammatica descritta si avrà la seguente dichiarazione:

```
<<SWE_Sensor_Request>> {  
  id_sensor = sensor1,  
  value_request = a2,  
  name = measurement,  
  optional = false }
```

Figura 5.8: Definizione della richiesta di un sensore usando la sintassi EBNF

In maniera del tutto analoga si specifica la Response (la risposta del sensore), supponendo ad esempio di voler intercettare i dati del sensore avente `id_sensor = sensor1` specificato in Figura 5.8, si avrà:

```
<<SWE_Sensor_Response>> {  
  id_sensor = sensor1,  
  value_response = a2,  
  update_data = name,  
  optional = true }
```

Figura 5.9: Definizione della risposta di un sensore usando la sintassi EBNF

I dati inseriti in queste dichiarazioni sono puramente a scopo informativo per mostrare il funzionamento della dichiarazione, ovviamente quando si andrà a descrivere il processo sarà necessario mettere gli identificativi delle attività coinvolte.

Queste dichiarazioni vanno fatte per tutti i sensori coinvolti nel processo BPMN, in modo da avere una panoramica completa tutti i dispositivi e dei dati che questi ultimi si scambiano tra di loro.

Capitolo 6

Applicazioni dell'estensione SWE_BPMN

L'autore della tesi ha applicato l'estensione proposta nel capitolo precedente ad una serie di casi di studio dove vengono descritti dei processi di business nel mondo reale che fanno uso di dispositivi IoT.

6.1 Supply chain

Un primo caso di studio preso in esame per l'applicazione dell'estensione SWE_BPMN ricavata utilizzando l'ontologia definita dal Open Geospatial Consortium (OGC) per lo scambio di dati tra sensori nel framework OGC SWE è quello della catena di approvvigionamento o supply chain in inglese.

Il processo inizia dalla swimlane Sensor Network e viene attivato ogni 60 secondi e vi sono una serie di attività (a1, a2, a3, a4, a5, a6, a7) che svolgono una serie di compiti come la misurazione della temperatura, l'invio dei dati al sistema backend, la generazione di messaggi d'errore e altre mansioni.

Ora si deve descrivere l'ambiente in cui tale processo lavora, ovvero definire tutti i dispositivi IoT presenti, in questo caso abbiamo dei sensori che si occupano di rilevare la temperatura e di inviare informazioni e dei tag RFID che sono dei piccoli chip, in cui sono contenute le informazioni riguardanti l'articolo o il prodotto che identifica, in questo caso i fiori (orchidee), il tutto viene codificato attraverso radiofrequenza (LF, HF o UHF).

Il diagramma BPMN del processo è stato riprodotto tramite il tool BPMN2 modeler di Eclipse che permette la creazione di diagrammi BPMN 2.0 e l'utilizzo di tutti i suoi elementi, permettendo anche la visione XML del processo di business e la modifica

diretta via codice, o anche la semplice visualizzazione tramite TextEditor ed è riportato in Figura 6.1.

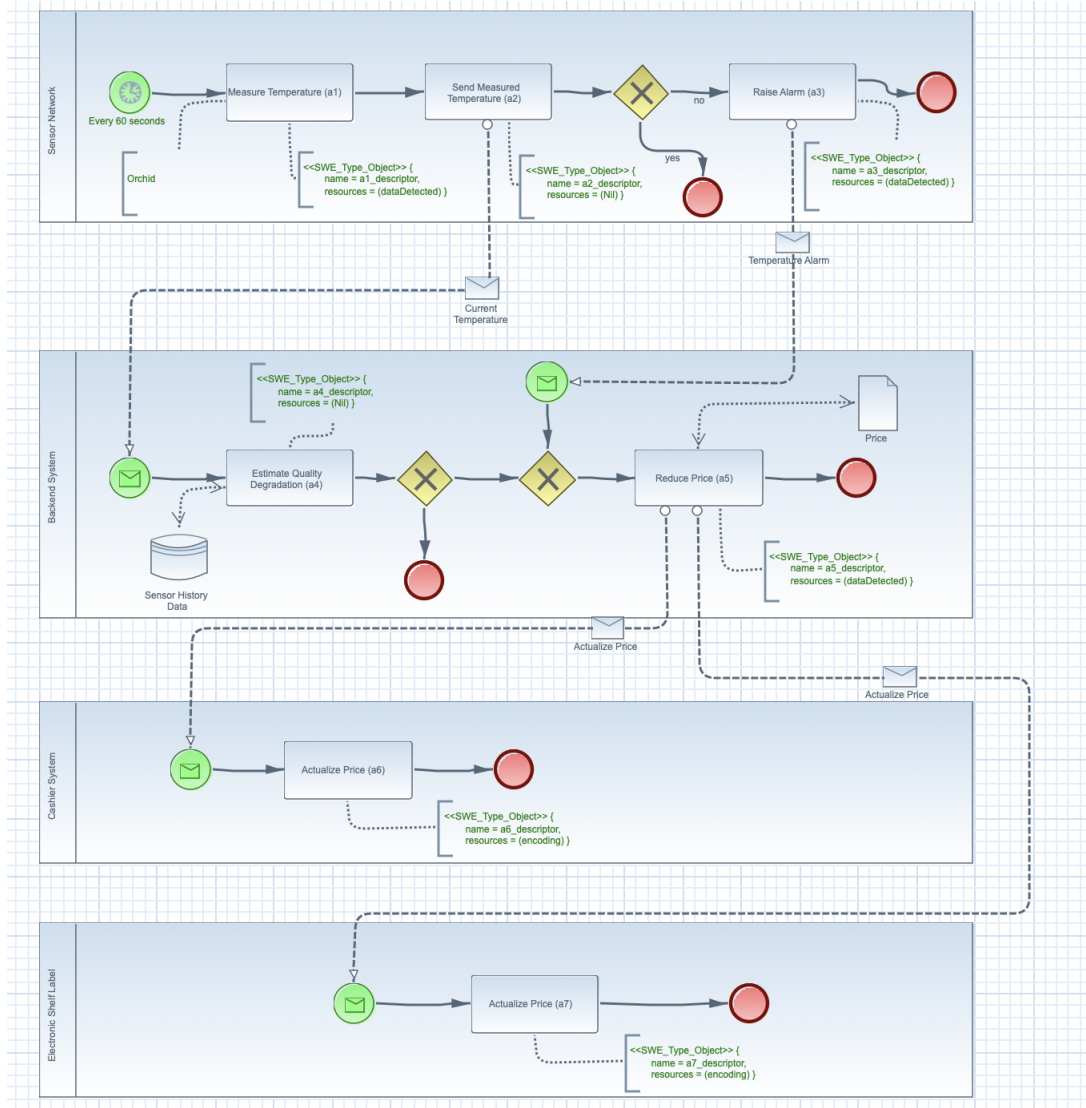


Figura 6.1: Diagramma BPMN della catena di approvvigionamento

Applicare l'estensione SWE_BPMN a questo caso di studio significa agganciare gli oggetti creati tramite la sintassi EBNF ai task BPMN e per far ciò si crea un elemento TextAnnotation (SWEAnnotation) associato al processo, tale meccanismo è riportato in figura 6.2, basta specificare la metaclassa che viene utilizzata ed i vari attributi definiti nel metamodello.

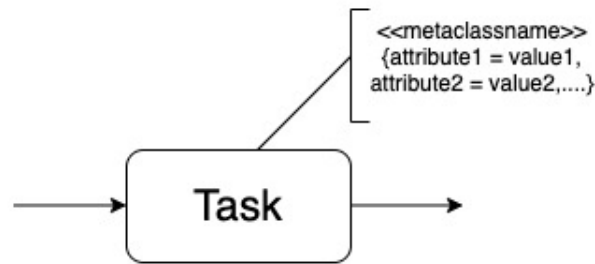


Figura 6.2: Processo di associazione di una TextAnnotation a un task BPMN

Dal punto di vista XML il tutto sarà visto come una TextAnnotation e per far sì che questo risulti valido occorre innanzitutto specificare il “name” del processo a cui vogliamo associarla e successivamente impostare un attributo “resources” in base all’elemento che il processo intende utilizzare. La vista XML di come avviene il tutto è riportata in Figura 6.3.

```

    <bpmn2:association id="Association_3" sourceRef="TextAnnotation_1" targetRef="Task_1"/>
  > <bpmn2:textAnnotation id="TextAnnotation_2">
    <bpmn2:text>
    &lt;&lt;SWE_Type_Object>> {
      name = a1_descriptor,
      resources = (dataDetected) }</bpmn2:text>
    </bpmn2:textAnnotation>
  > <bpmn2:textAnnotation id="TextAnnotation_3">
    <bpmn2:text>
    &lt;&lt;SWE_Type_Object>> {
      name = a2_descriptor,
      resources = (Nil) }</bpmn2:text>
    </bpmn2:textAnnotation>
  > <bpmn2:textAnnotation id="TextAnnotation_4">
    <bpmn2:text>
    &lt;&lt;SWE_Type_Object>> {
      name = a3_descriptor,
      resources = (dataDetected) }</bpmn2:text>
    </bpmn2:textAnnotation>
  > <bpmn2:association id="Association_4" sourceRef="TextAnnotation_2" targetRef="Task_1"/>
    <bpmn2:association id="Association_5" sourceRef="TextAnnotation_3" targetRef="Task_2"/>
    <bpmn2:association id="Association_6" sourceRef="TextAnnotation_4" targetRef="Task_3"/>

```

Figura 6.3: Visione XML delle TextAnnotation

Come si può vedere dalla Figura 6.3 una TextAnnotation viene associata al task tramite l’attributo “targetRef” nel quale viene riportato appunto l’identificativo del processo associato.

Nell’attività a1 si va ad associare quella determinata TextAnnotation perché si va a misurare la temperatura di un oggetto fisico (orchidea) e quindi è necessario avere un dato reale, perciò si definisce l’oggetto SWE_Type.Object sfruttando la metaclassa SWE_Data_Type dell’estensione che permette appunto di intercettare i dati provenienti dai sensori.

Nell’attività a2 vengono inviati dei dati e quindi vi è uno scambio di informazioni tra i vari dispositivi IoT, per questo motivo si va a definire sempre l’oggetto

SWE_Type_Object ma ora si sfrutta la metaclassa SWE_Simple dell'estensione che permette di andare a “misurare” lo stream dei dati che vengono inviati durante l'esecuzione del processo.

La terza metaclassa dell'estensione viene utilizzata dalle attività a6 e a7 e la visione XML della TextAnnotation è riportata in Figura 6.4.

```
<bpmn2:textAnnotation id="TextAnnotation_7">
  <bpmn2:text>
    &lt;&lt;SWE_Type_Object>> {
      name = a6_descriptor,
      resources = (encoding) }</bpmn2:text>
    </bpmn2:text>
  </bpmn2:textAnnotation>
  <bpmn2:association id="Association_9" sourceRef="TextAnnotation_7" targetRef="Task_6"/>
```

Figura 6.4: Visione XML della metaclassa SWE_Encode

Queste attività hanno il compito di andare ad aggiornare il prezzo dell'oggetto per far ciò devono codificare i dati ricevuti in modo che non siano in chiaro durante la trasmissione e successivamente far terminare il processo.

Tutte le altre attività svolgono mansioni simili a quelle già descritte pertanto andranno ad utilizzare le tre metaclassi dell'estensione in base alle loro esigenze durante l'esecuzione del processo di business.

6.2 Flower distribution

Un secondo caso di studio preso in esame per l'applicazione dell'estensione proposta SWE_BPMN riguarda il processo di distribuzione e consegna di fiori.

Il modello di business rappresenta la procedura di classificazione, conservazione e distribuzione del prodotto floreale all'interno di un'azienda. Il processo inizia con un evento scatenante che indica che alcuni fiori freschi sono stati appena trasportati in un punto di raccolta e sono pronti per la distribuzione. All'interno di ogni fioriera sono installati sensori ambientali per monitorare la temperatura e l'umidità.

I tag RFID possono anche essere applicati al contenitore per segnalare l'arrivo dei contenitori e per registrare il tempo impiegato per il trasporto dei fiori. Le informazioni di cui sopra vengono utilizzate per creare un set di dati sulla condizione dei fiori. Questo set di dati verrà utilizzato per classificare i fiori in base alla loro condizione. Successivamente verranno distribuiti fiori di buona qualità ai rivenditori locali e venduti come fiori freschi. I fiori di qualità mediocre saranno distribuiti ai magazzini per lo stoccaggio e potranno essere trasformati in prodotti floreali secchi. Diverse categorie di fiori avranno requisiti diversi sui magazzini in termini di temperatura, umidità, posizione e costo richiesti.

Anche per questo caso di studio il diagramma BPMN del processo è stato riprodotto tramite il tool BPMN2 modeler di Eclipse ed è riportato in Figura 6.5.

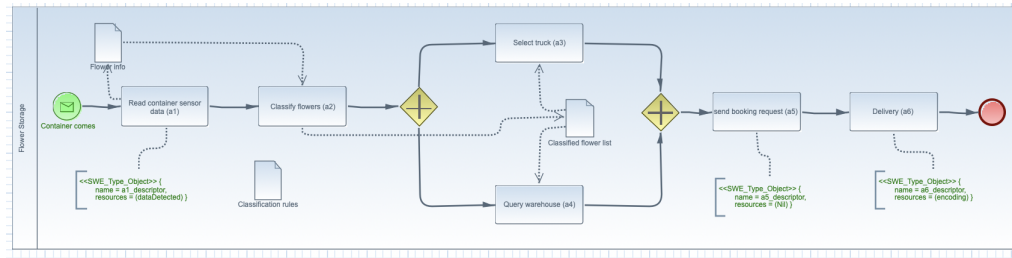


Figura 6.5: Diagramma BPMN Flower distribution

Analogamente al primo caso di studio, anche in questa situazione applicare l'estensione SWE_BPMN significa agganciare gli oggetti creati tramite la sintassi EBNF ai task BPMN e per far ciò si crea un elemento TextAnnotation associato al processo, ovviamente dal punto di vista XML il tutto sarà visto come una TextAnnotation, la situazione è riportata in Figura 6.6.

```
<bpmn2:textAnnotation id="TextAnnotation_1">
  <bpmn2:text>
    &lt;&lt;SWE_Type_Object>> {
      name = a1_descriptor,
      resources = (dataDetected) }&lt;/bpmn2:text>
    </bpmn2:textAnnotation>
    <bpmn2:association id="Association_1" sourceRef="TextAnnotation_1" targetRef="Task_1"/>
    <bpmn2:textAnnotation id="TextAnnotation_2">
      <bpmn2:text>
        &lt;&lt;SWE_Type_Object>> {
          name = a5_descriptor,
          resources = (Nil) }&lt;/bpmn2:text>
        </bpmn2:textAnnotation>
        <bpmn2:association id="Association_2" sourceRef="TextAnnotation_2" targetRef="Task_5"/>
        <bpmn2:textAnnotation id="TextAnnotation_3">
          <bpmn2:text>
            &lt;&lt;SWE_Type_Object>> {
              name = a6_descriptor,
              resources = (encoding) }&lt;/bpmn2:text>
            </bpmn2:textAnnotation>
            <bpmn2:association id="Association_3" sourceRef="TextAnnotation_3" targetRef="Task_6"/>
          </bpmn2:textAnnotation>
        </bpmn2:association>
      </bpmn2:association>
    </bpmn2:association>
  </bpmn2:association>
</bpmn2:association>
```

Figura 6.6: Visione XML delle TextAnnotation Flower distribution

Si nota che le metaclassi utilizzate sono le stesse utilizzate nel primo caso di studio e questo è dovuto grazie alla semplicità di applicazione dell'estensione SWE_BPMN che rende il metamodello sempre valido.

In questa situazione le TextAnnotation non vengono agganciate a tutti i task presenti nel diagramma ma solo a quelli a cui è necessario.

Ad esempio, l'attività a1 va leggere i dati dei sensori e quindi una volta definito l'oggetto SWE_Type_Object si va a sfruttare la metaclassa SWE_Data_Type che come descritto precedentemente permette di intercettare i dati rivelati dai sensori. L'attività a6 invia la richiesta e quindi i dati vengono inviati, si sfrutta la metaclassa SWE_Simple dell'estensione che controlla il flusso di dati inviati e/o scambiati tra i sensori e l'attività

a7 invia gli oggetti e quindi i dati per questioni di sicurezza vanno codificati e non lasciati in chiaro.

6.3 Vantaggi dell'estensione

L'estensione SWE_BPMN risulta essere semplice da applicare e allo stesso tempo funzionante a questi casi di studio grazie alla natura stessa dell'estensione eseguita, poichè si va ad estendere il metamodello BPMN sfruttando il BaseElement dal quale derivano tutti i suoi elementi.

Successivamente una volta definita una grammatica generale utilizzando una sintassi EBNF si possono agganciare le metaclassi ricavate dall'ontologia agli oggetti BPMN mantenendo quindi l'estensione valida e allo stesso tempo il metamodello funzionante.

Per questo motivo l'estensione proposta consente l'analisi delle prestazioni e dell'affidabilità dei processi di business in fase di progettazione, valutando l'impatto di ogni singolo data object che verrà elaborato/mandato da un sensore. Non deve passare in secondo piano l'aspetto relativo all'utilizzo di questi sensori e tag RFID poichè ogni volta che si rappresenta un ambiente in cui un determinato processo di business opera è fondamentale descrivere e definire utilizzando una sintassi generale, in questo caso quella EBNF, tutti i dispositivi IoT presenti in modo da avere una panoramica completa della situazione che si ha davanti.

Capitolo 7

Processi di business che non usano IoT

L'autore della tesi per motivare l'utilizzo dell'estensione SWE_BPMN ha preso in esame una ricerca fatta su un processo di business [6] che non fa uso di dispositivi IoT durante lo svolgimento del processo, mostrando che l'estensione in tale situazione sarebbe stata molto efficace.

7.1 Remote Healthcare monitor system

Il moderno sistema sanitario combina diverse entità come istituzioni, operatori sanitari, informazioni sui pazienti e al fine di soddisfare le esigenze individuali dei diversi ruoli è fondamentale una buona comprensione dell'intero processo.

Pertanto, è necessario un linguaggio di modellazione strutturato per visualizzare il concetto di sistema sanitario, permettendo inoltre la modellazione di tali processi al fine di aiutare a progettare e implementare il sistema informativo.

Il BPMN viene incontro a queste esigenze, permettendo quindi di contenere enormi comunicazioni e interazioni tra pazienti, servizio fornitori, partner e i diversi dipartimenti.

Oggi è abbastanza comune accedere alle informazioni sanitarie online, come ad esempio la cartella clinica elettronica (EHR), questa digitalizzazione comporta due aspetti fondamentali:

- la facilità nel fruire le informazioni, come ad esempio analisi o risultati di esami specifici da parte dei pazienti;
- un incremento della sicurezza dei dati sensibili dei pazienti.

Tradizionalmente, i requisiti di sicurezza sono considerati solo dopo la definizione del processo di business e BPMN attualmente non supporta aspetti come la riservatezza, l'integrità e la disponibilità in modo adeguato.

Un esempio di diagramma BPMN per il sistema di monitoraggio sanitario a distanza (remote healthcare monitor system) è riportato in Figura 7.1. Il diagramma in questione è stato riprodotto usando il tool di modellazione di Eclipse BPMN2 modeler.

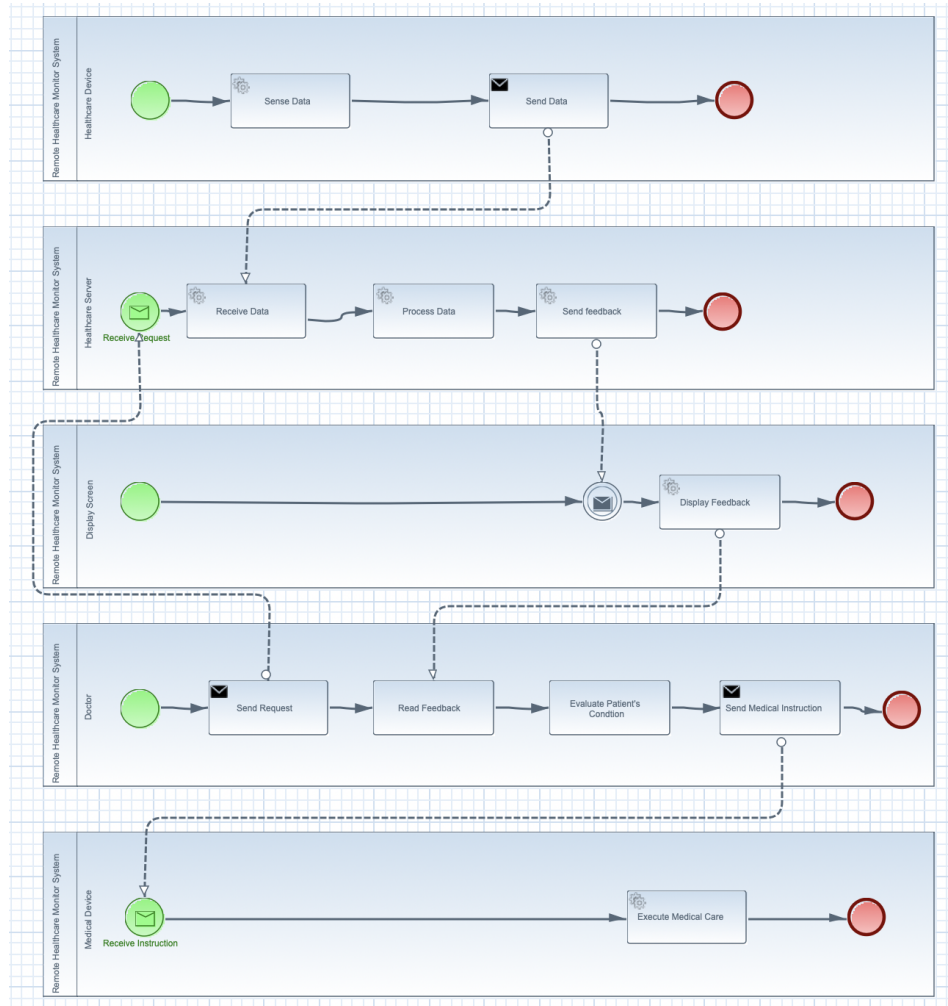


Figura 7.1: Diagramma BPMN di un processo sanitario a distanza

L'idea della ricerca è quella di definire i requisiti di sicurezza a livello di astrazione, fornendo una serie di nuove estensioni progettate in BPMN per modellare i concetti di sicurezza nel processo sanitario. Queste nuove estensioni, chiamate notazioni di sicurezza, sono in grado di lavorare insieme alle notazioni BPMN esistenti e apportano diverse nuove funzionalità per visualizzare i requisiti di sicurezza.

Le notazioni estese mirano a rappresentare i concetti della sicurezza piuttosto che la tecnologia utilizzata per attuarla, ignorando i meccanismi tecnici utilizzati per raggiungere i requisiti necessari per avere un processo sicuro.

Queste estensioni sono ottenute creando degli elementi dell'estensione che rappresentano la sicurezza del processo come ad esempio l'autenticazione, il controllo,

autorizzazioni e altre. Lo schema di tutti gli elementi è riportato in Figura 7.2.









Element	Type	Design
Security Task	Task	
Authentication	Boundary Event	
Access Control	Boundary Event	
Authorization	Boundary Event	
Harm Protection	Boundary Event	
Encrypted Message	Intermediate Event	
Non Repudiation	Intermediate Event	
Secure Communication	Intermediate Event	

Figura 7.2: Elementi estesi per i requisiti di sicurezza

Dal punto di vista teorico e progettuale la scelta di attuare questo tipo di estensione può risultare corretta, poiché si va ad evidenziare una delle proprietà non funzionali del BPMN, in questo caso la sicurezza.

Tuttavia l'estensione risulta essere poco ragionevole se la si guarda a livello di sicurezza soprattutto a livello dei dati sensibili dei pazienti. Nonostante siano presenti degli elementi che attuano la codifica e la decodifica non vi è alcuna traccia dei metodi usati per codificare tali dati (metodi di encoding) rendendo il tutto troppo teorico e poco realizzabile a livello pratico.

Per questo motivo all'interno di questo processo di business è possibile utilizzare l'estensione SWE_BPMN proposta dall'autore della tesi, andando a creare una rete di dispositivi IoT come sensori e tag RFID che contengono le varie informazioni dei pazienti e al momento opportuno le inviano verso coloro che li hanno richiesti, quindi i dati viaggieranno sì all'interno della rete ma saranno comunque sempre protetti.

7.2 SWE_BPMN nel processo sanitario

Una possibile soluzione per utilizzare l'estensione SWE_BPMN in questo processo di business è quello di associare ad ogni paziente un tag RFID contenente l'identificativo del paziente e usare dei sensori che si occupano di captare i parametri dei pazienti (temperatura, pressione sanguigna, attività, ecc.) e poi utilizzarli per codificare questi dati che saranno inviati nei nodi della rete creata.

Pertanto, si deve andare a descrivere l'ambiente in cui il processo lavora, ovvero occorre dichiarare tramite la sintassi EBNF i dispositivi IoT che operano, ne basta una generale che poi andrà bene per tutti gli altri.

Un sensore come descritto nel SWE Service Model affinché possa essere descritto ha bisogno che vengano definite sia la richiesta (request) che la risposta (response). La richiesta è riportata in Figura 7.3.

```
<<SWE_Sensor_Resquest>> {  
  id_sensor = sensor1,  
  value_request = sendTask,  
  name = "send data",  
  optional = false }
```

Figura 7.3: Definizione sensore per invio dati

In maniera del tutto analoga si va a definire la risposta del sensore attenendosi alla sintassi del SWE Service Model.

```
<<SWE_Sensor_Response>> {  
  id_sensor = sensor1,  
  value_request = receiveTask,  
  name = "receive data",  
  optional = true }
```

Figura 7.4: Definizione sensore per ricezione dati

Quindi una volta definito l'ambiente in cui il processo opera per applicare l'estensione SWE_BPMN bisogna agganciare gli oggetti creati tramite la sintassi EBNF ai task BPMN e creare quindi un elemento SWEAnnotation associato al processo, come descritto nel Capitolo 5.

L'estensione SWE_BPMN verrà applicata in tre task che si occupano dei dati sensibili dei pazienti all'interno del processo sanitario ed è mostrata in Figura 7.5.

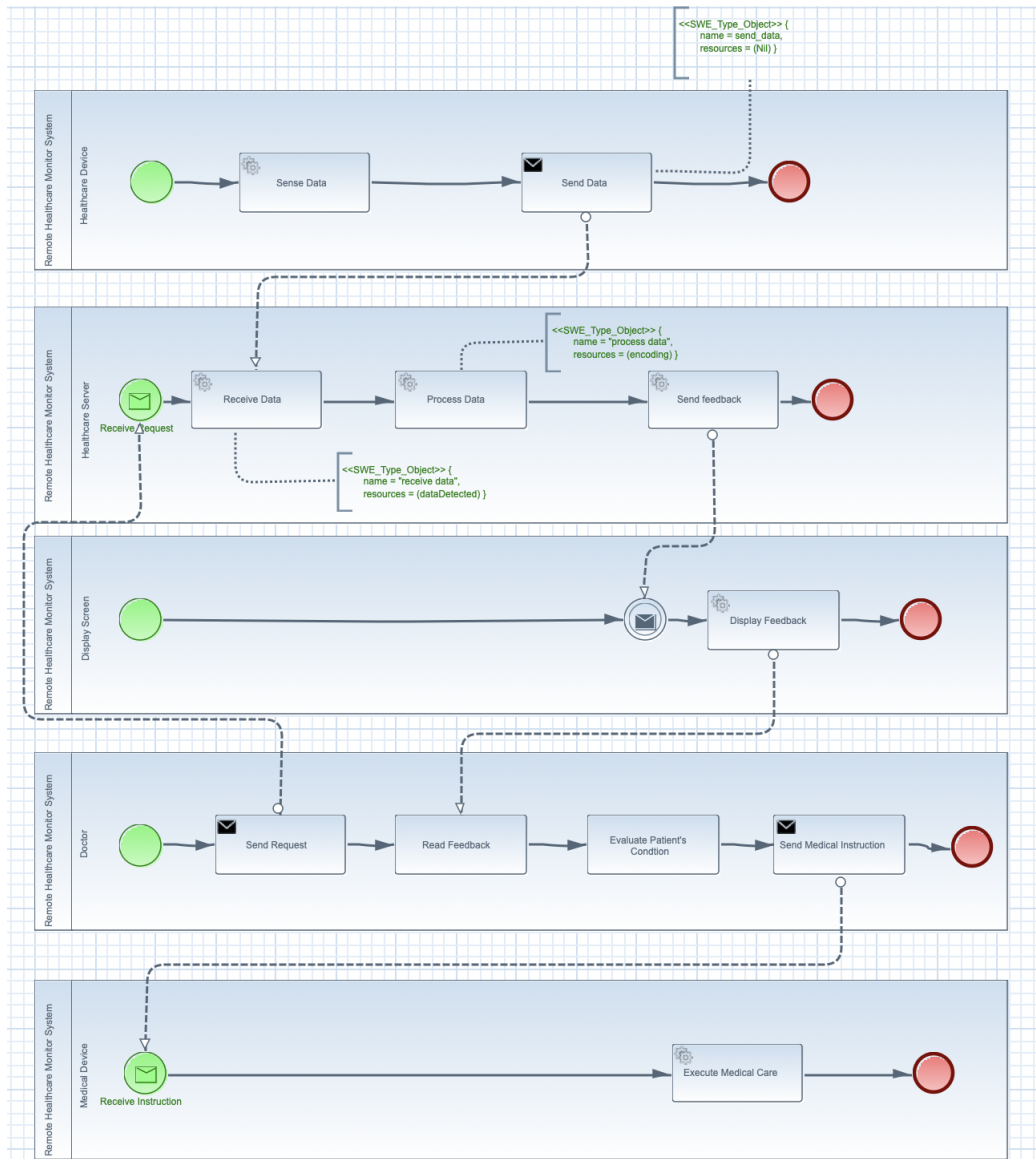


Figura 7.5: Applicazione dell'estensione SWE BPMN al processo di business

Applicare le SWEAnnotation a questi task BPMN significa agganciare gli oggetti del metamodello dell'estensione ovvero le varie metaclassi definite in SWE_BPMN. La visione XML delle SWEAnnotation applicate a questo processo di business è riportata in Figura 7.6 e si adattano perfettamente al diagramma del processo.

L'estensione viene applicata ai task nel processo di business che trattano i dati sensibili dei pazienti (invio, ricezione ed elaborazione). Nel task "Send Data" si va a definire l'oggetto SWE.Type.Object e dato che si vanno ad inviare dei dati si frutta la metaclass SWE.Simple che ci permette di controllare lo stream delle informazioni che vengono inviate.

```
<bpmn2:textAnnotation id="TextAnnotation_1">
  <bpmn2:text>
    &lt;&lt;SWE_Type_Object>> {
      name = send_data,
      resources = (Nil) }
    </bpmn2:text>
  </bpmn2:textAnnotation>
  <bpmn2:association id="Association_1" sourceRef="TextAnnotation_1" targetRef="SendTask_1"/>
</bpmn2:process>
```

Figura 7.6: Visione XML delle TextAnnotation nel processo sanitario

Dopodiché tramite un message flow i dati che vengono inviati al task "Receive Data", si definisce l'oggetto dell'estensione SWE_Type_Object solo che questa volta si sfrutta la metaclassa SWE_Data_Type la quale permette di gestire i dati che vengono ricevuti/intercettati durante il processo.

A questo punto i dati vengono elaborati, trattandosi di dati sensibili questi vengono codificati e successivamente decodificati per poter essere letti dal ricevente, quindi si definisce sempre l'oggetto SWE_Type_Object ma ora si sfrutta la metaclassa SWE_Encode utilizzando uno dei metodi disponibili per la codifica (TextEncoding, BinaryEncoding, XMLEncoding).

L'estensione SWE_BPMN permette di evidenziare una delle proprietà non funzionali del BPMN come la sicurezza, permettendone la realizzazione attraverso l'utilizzo di dispositivi IoT che la mettono in atto senza passare attraverso dei task interni al processo che potrebbero rendere il sistema poco efficiente.

7.3 Considerazioni finali

L'idea di modellare il funzionamento di un sistema di monitoraggio sanitario a distanza utilizzando il BPMN è molto valida, soprattutto grazie alla grande espressività, semplicità e ricchezza semantica di tale linguaggio.

Dal punto di vista progettuale, rappresentare una proprietà non funzionale del BPMN attraverso dei task specifici che colmano la lacuna della sicurezza sia dei dati che dell'esecuzione del processo stesso va bene, poiché molti aspetti del linguaggio di modellazione vengono trattati in maniera astratta e non formale (ad esempio la scelta di molti simboli è lasciata al progettista).

Se si guarda a questa soluzione da un punto di vista pratico si può evincere che non è una scelta congeniale utilizzare questo tipo di approccio e quindi l'utilizzo dell'estensione SWE_BPMN viene incontro a queste esigenze.

Data la digitalizzazione di tutte le operazioni in questi ultimi anni e l'incremento nell'utilizzo di apparecchi intelligenti, usare dei dispositivi IoT per monitorare i pazienti sarebbe molto più comodo sia a livello di organizzazione che per il trattamento dei dati sensibili, considerando anche il basso costo dei tag RFID e dei sensori.

I dati sono delle informazioni private e che vanno protette da qualsiasi attacco e l'estensione proposta dall'autore della tesi lo permette ed è possibile utilizzarla senza stravolgere l'idea dell'intero processo di business.

Il punto di forza dell'estensione SWE_BPMN è proprio nell'utilizzo di dispositivi di IoT nei vari processi di business, in modo tale da rendere quest'ultimi più attuali e permettendo alle società di svolgere le proprie attività in maniera più semplificata.

L'estensione rende il metamodello BPMN valido e sempre funzionante poiché è derivata dal BaseElement, ovvero la classe da dove provengono tutti gli elementi del linguaggio di modellazione e tramite una TextAnnotation (SWEAnnotation) l'estensione viene messa in pratica in modo efficiente e sicuro, andando quindi ad esprimere in maniera chiara e dettagliata le varie proprietà non funzionali del linguaggio di modellazione BPMN.

Riferimenti

- [1] Maria Bermudez-Edo, Tarek Elsaleh, Payam Barnaghi, and Kerry Taylor. Iot-lite: A lightweight semantic model for the internet of things. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDDCom/IoP/SmartWorld)*, pages 90–97, 2016.
- [2] Paolo Bocciarelli, Andrea D’Ambrogio, Andrea Giglio, and Emiliano Paglia. A bpmn extension to enable the explicit modeling of task resources. In *Py_BPMN*, 11 2016.
- [3] Paolo Bocciarelli, Andrea D’Ambrogio, Andrea Giglio, and Emiliano Paglia. A bpmn extension for modeling cyber-physical-production-systems in the context of industry 4.0. *IEE Xplore*, pages 599–604, 05 2017.
- [4] Johannes Echterhoff. Opengis® swe service model implementation standard. *OGC*, pages 1–156, 01 2011.
- [5] Alexandre Robin. Ogc® swe common data model encoding standard. *OGC*, pages 1–207, 01 2011.
- [6] Koh Song Sang and Bo Zhou. Bpmn security extensions for healthcare process. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 2340–2345, 2015.

- [7] Kunal Suri, Walid Gaaloul, Arnaud Cuccuru, and Sébastien Gérard. Semantic framework for internet of things-aware business process development. *IEE Xplore*, 06 2017.
- [8] Shulong Wang, Yibin Hou, Fang Gao, and Songsong Ma. Ontology-based resource description model for internet of things. In *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 105–108, 2016.
- [9] Mathias Weske. *Business Process Management*. Springer, 2012.
- [10] Karim Zarour, Djamel Benmerzoug, Nawal Guermouche, and Khalil Drira. A systematic literature review on bpmn extensions. *Business Process Management Journal*, ahead-of-print, 11 2019.

Ringraziamenti

Sono giunto alla fine di un percorso molto importante e formativo della mia persona. Questo elaborato rappresenta la conclusione di un cammino che non mi è stato facile portare al compimento.

Intendo ringraziare in queste poche righe colui che ha pensato più volte di mollare tutto, ma puntualmente si è rimboccato le maniche ed ha ripreso da dove aveva lasciato ogni volta, ovvero me stesso. Sono molto testardo ed autocritico nei miei confronti ma se non fossi stato così ora non sarei arrivato a questa nuova fase della mia vita.

Ringrazio infinitamente i miei genitori che mi hanno sempre sostenuto, appoggiando ogni mia decisione, fin dalla scelta del mio percorso di studi e soprattutto nell'avermi tirato su nei momenti più difficili.

Un grazie a tutti i colleghi universitari che ho incontrato durante questa esperienza e con i quali ho condiviso questi anni.

Infine, ringrazio l'università e il mio relatore Andrea D'Ambrogio, che mi hanno accompagnato in questo ultimo traguardo permettendomi di realizzare questo sogno che inizialmente mi sembrava molto lontano.