

Università degli studi di Milano-Bicocca

Architettura del Software

Autori:

Alberto Gibertini - 816069 - a.gibertini@campus.unimib.com

Thomas Albericci - 816142 - t.albericci1@campus.unimib.com



Note sul pdf

- Le immagini sono in formato SVG (vettoriale), quindi se vengono zoomate non si sgranano. Per visibilità i diagrammi vengono divisi e analizzati singolarmente.

1. Introduzione

Il progetto riguarda la realizzazione di un sistema di gestione operativa, che controlla centraline di distribuzione di energia elettrica.

Il sistema si occupa di acquisire dati in tempo reale da ogni centralina e di controllare se questi dati presentino delle anomalie.

Nel caso di situazioni anomale, il sistema contatta il servizio tecnico centrale, che a sua volta contatta un operatore per risolvere l'anomalia.

2. Architettura Del Problema

In questo capitolo vengono analizzati gli attori del problema, le entità e i casi d'uso.

Gli attori identificati sono:

- Sensore: un sensore vero e proprio in grado di rilevare un dato, localizzato nella centralina.
- Controller: una parte della centralina in grado di effettuare operazioni quali la verifica di una anomalia, la disattivazione della centralina e la creazione di una nuova policy per la centralina stessa.
- Servizio tecnico: ha luogo in una sede esterna, riceve le anomalie dai controller delle centraline e si occupa di identificare l'operatore migliore per una determinata centralina e notificarlo.
- Operatore: si occupa di comunicare l'inizio dell'attività di ripristino, di riattivare la centralina e di comunicare la fine dell'attività di ripristino.

Diagramma dei casi d'uso

In questo diagramma vengono identificati gli **attori** e i loro **casi d'uso**.

L'**attore** rappresenta il ruolo che un'entità esterna assume quando interagisce con il sistema.

Il **caso d'uso** specifica una interazione tra il sistema e uno o più attori.

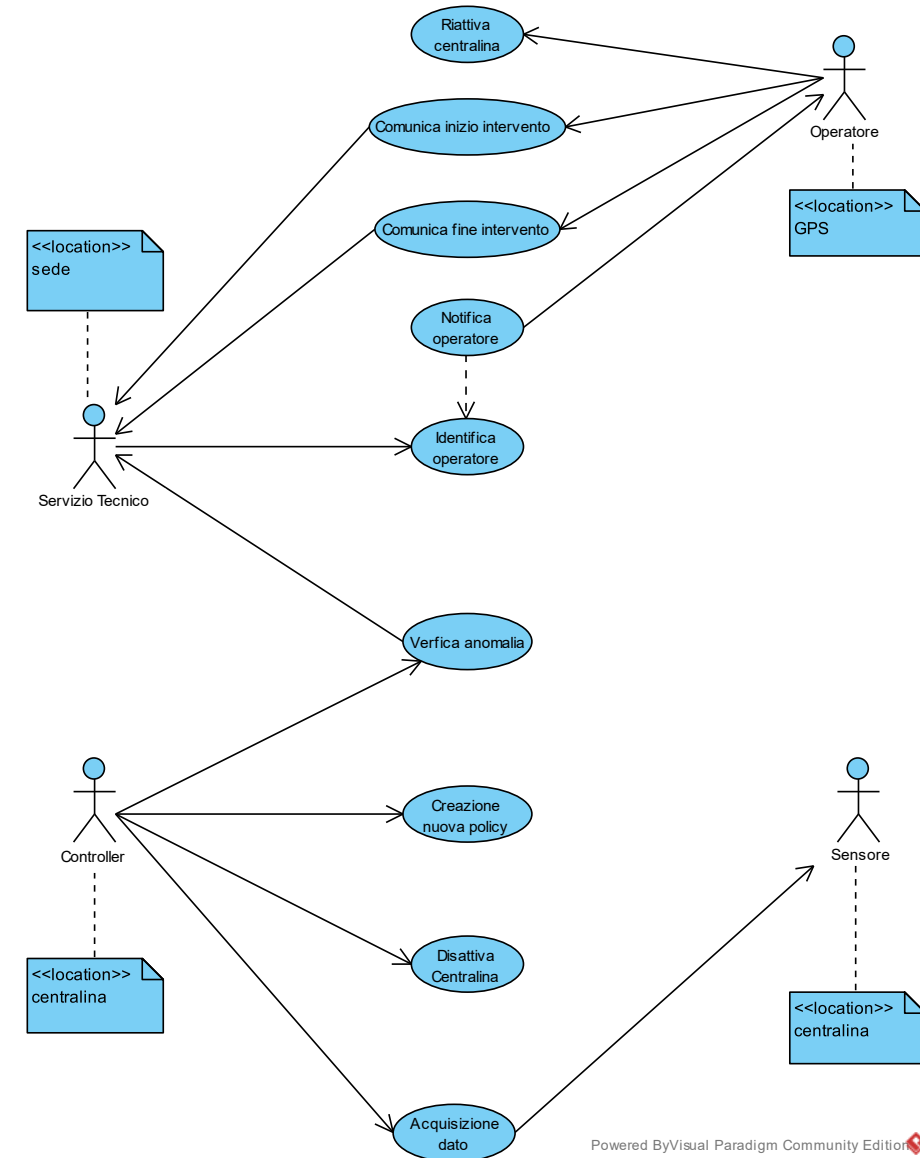


Diagramma delle classi

Questo diagramma descrive le classi e le relazioni tra di esse.

Individua quali sono gli elementi del sistema.

Quando si verifica una anomalia, viene creato un oggetto **AnomaliaDellaCentralina**, contenente l'id della centralina e l'id del dato del sensore anomalo, dal quale è possibile accedere all'oggetto **Centralina** e all'oggetto **DatoDelSensore**. **AnomaliaDellaCentralina** serve al servizio tecnico per individuare l'operatore adatto.

L'oggetto **NotificaAnomaliaOperatore** contiene le informazioni per ripristinare una centralina, tra cui è presente anche l'id dell'oggetto **AnomaliaDellaCentralina**, che gli permette di sapere quale centralina deve essere riparata.

L'operatore durante il suo intervento comunica l'inizio e la fine, tramite i due oggetti **ComunicazioneInizio** e **ComunicazioneFine**.

Alla centralina viene assegnato un oggetto **Policy**, contenente le regole che la centralina deve seguire.

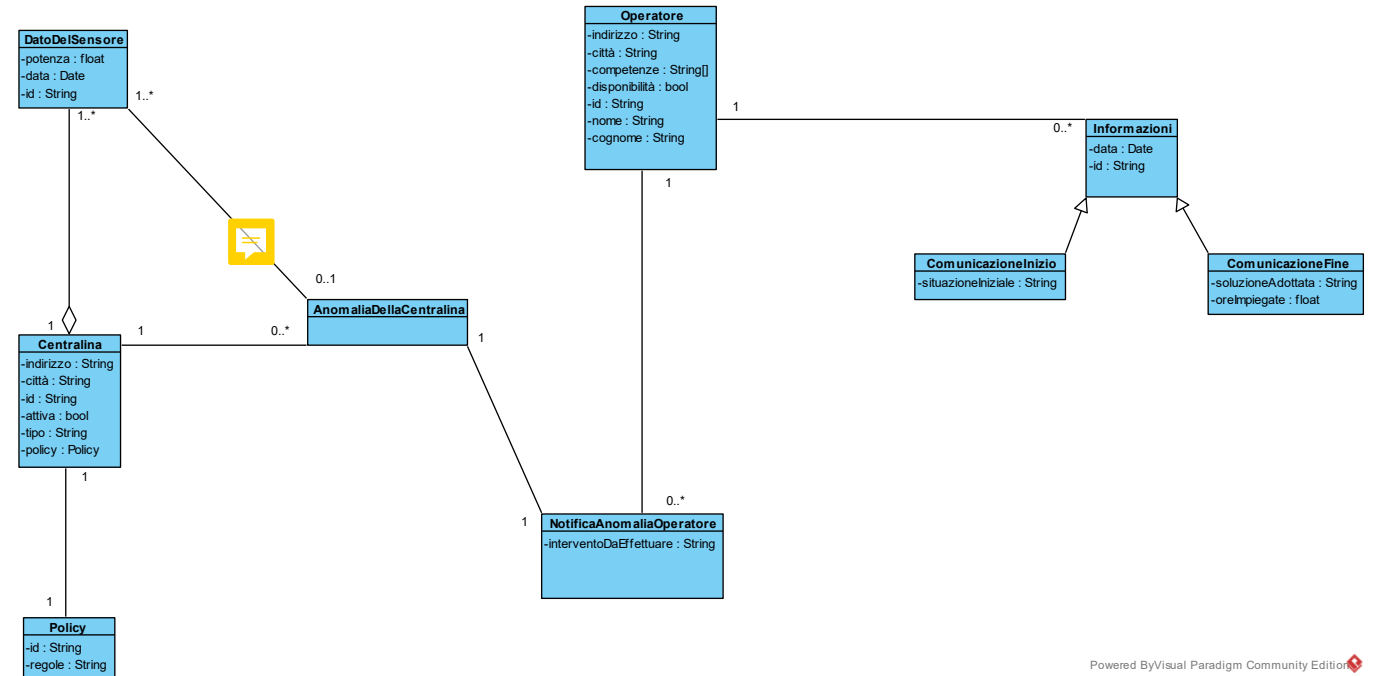
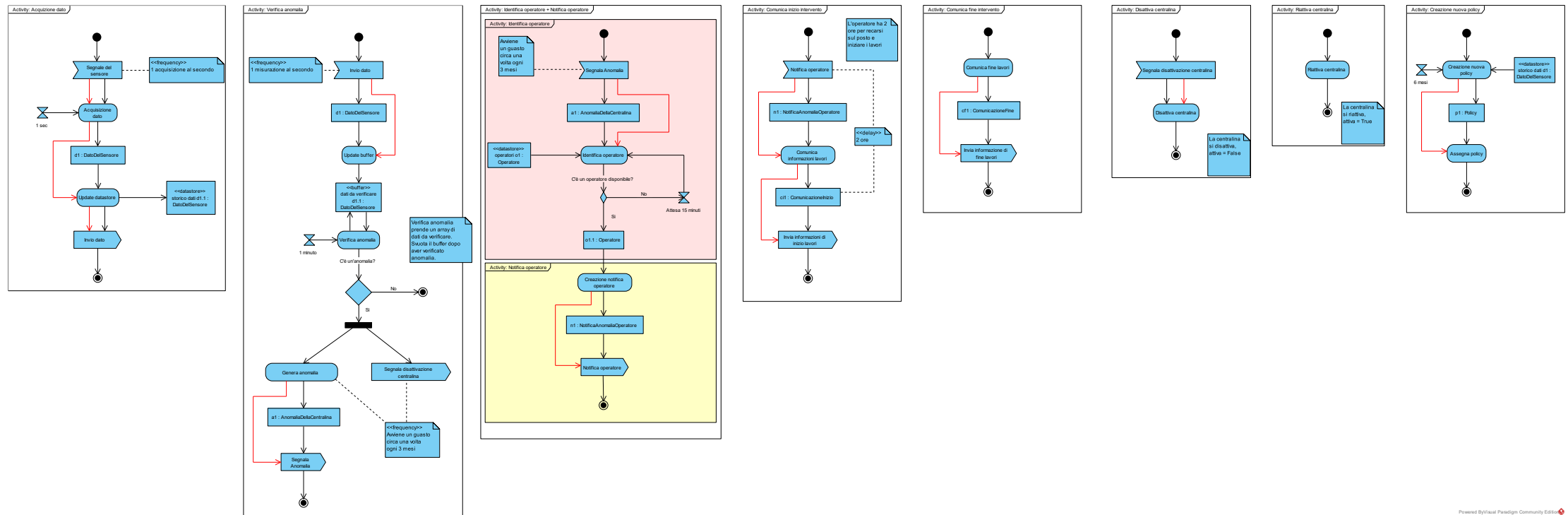


Diagramma delle attività

Partendo dal diagramma dei casi d'uso, viene modellata una attività per ogni caso d'uso, sono state create 8 attività:

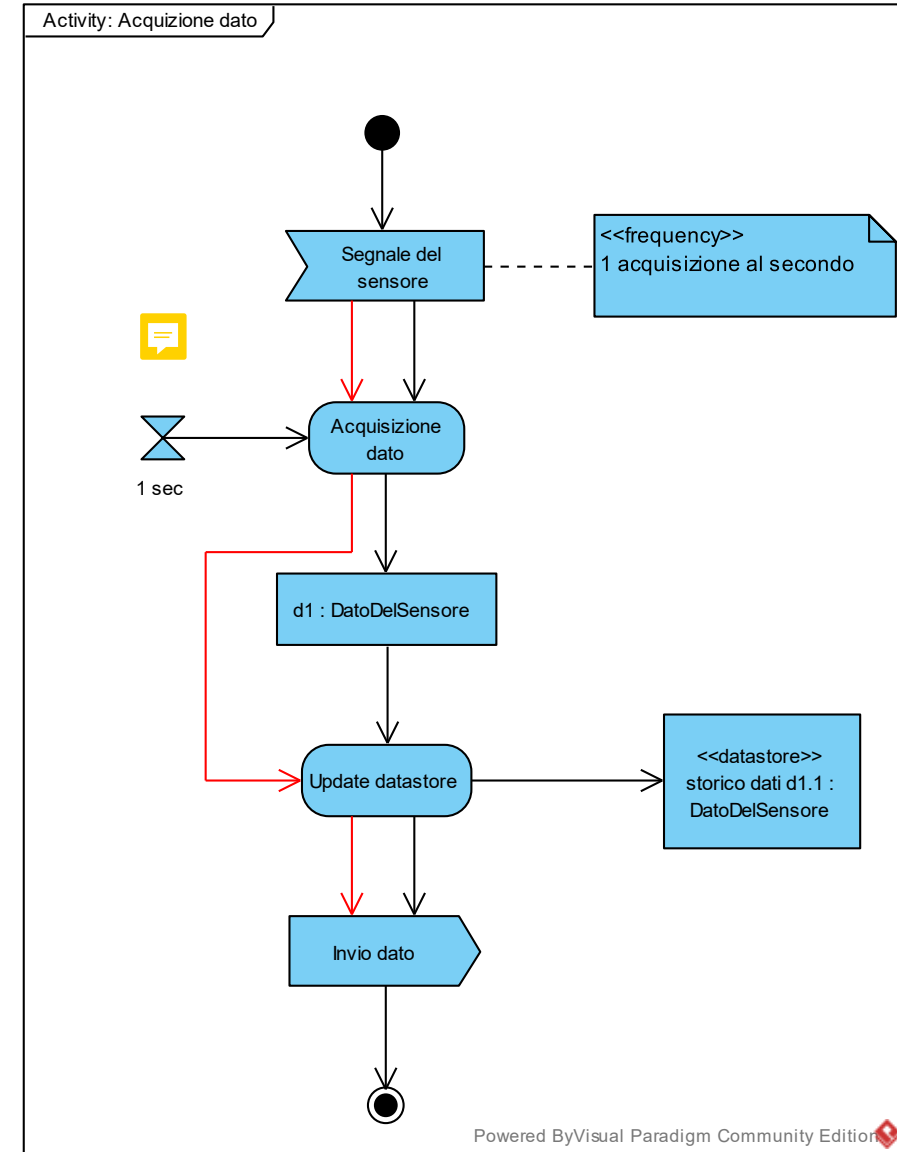
- Acquisizione dato
- Verifica anomalia
- Identifica operatore + Notifica operatore
- Comunica inizio intervento
- Comunica fine intervento
- Disattiva centralina
- Riattiva centralina
- Creazione nuova policy

Diagramma delle attività



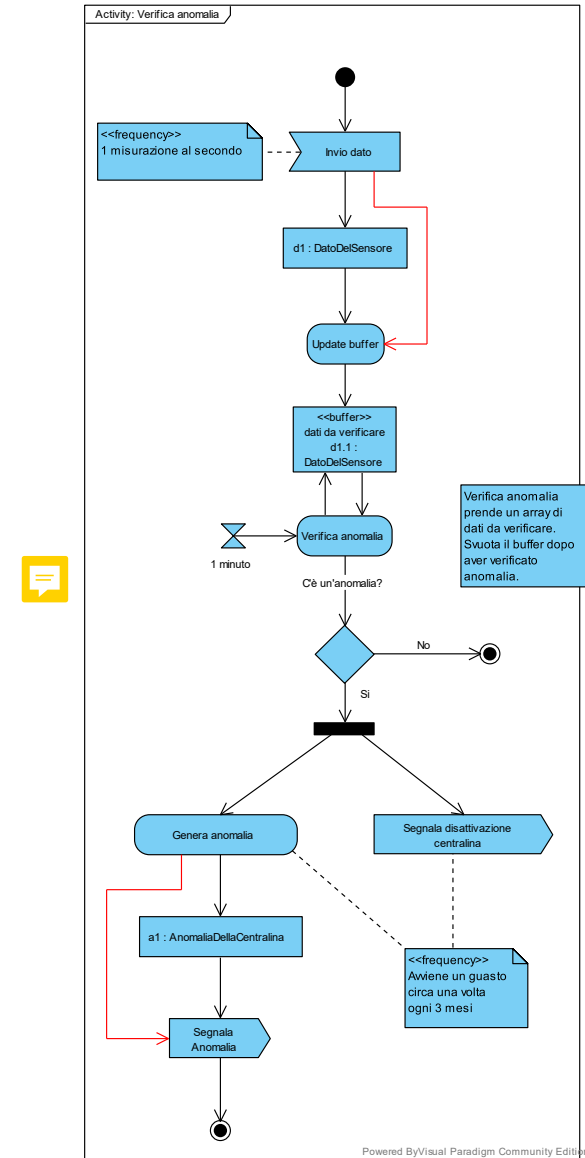
Acquisizione dato

- Una volta al secondo viene acquisito un dato dal sensore.
- Il dato viene inserito nel datastore Storico dati, che è associato ad una singola centralina.
- Il dato viene infine inviato all'activity Verifica anomalia.



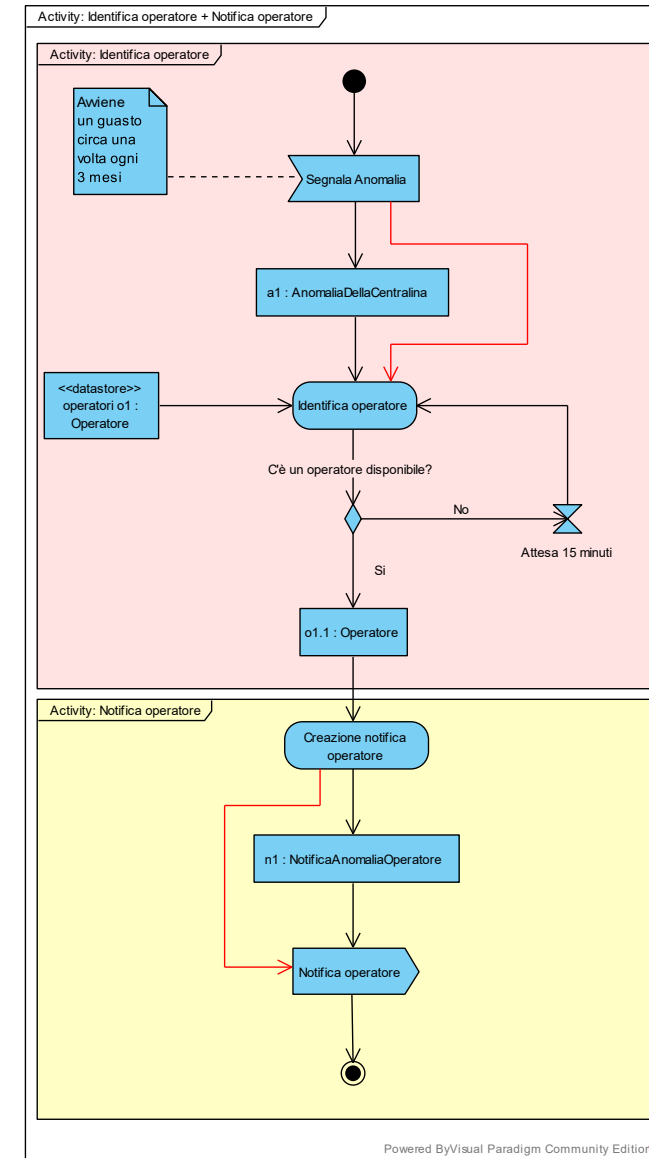
Verifica anomalia

- Riceve il dato dall'activity Acquisizione dato.
- Il dato viene inserito in un buffer, chiamato Dati da verificare. Il buffer è associato ad una singola centralina
- Una volta al minuto viene eseguita l'operazione Verifica anomalia, che prende in input un array di dati dal buffer, lo svuota e controlla se si sia verificata una anomalia.
 - Nel caso si fosse verificata una anomalia, in parallelo:
 - Viene inviato un segnale di disattivazione centralina all'activity Disattiva centralina
 - Viene generato un oggetto AnomaliaDellaCentralina e viene inviato all'activity Identifica Operatore
 - Nel caso non si fosse verificata una anomalia, l'activity termina.



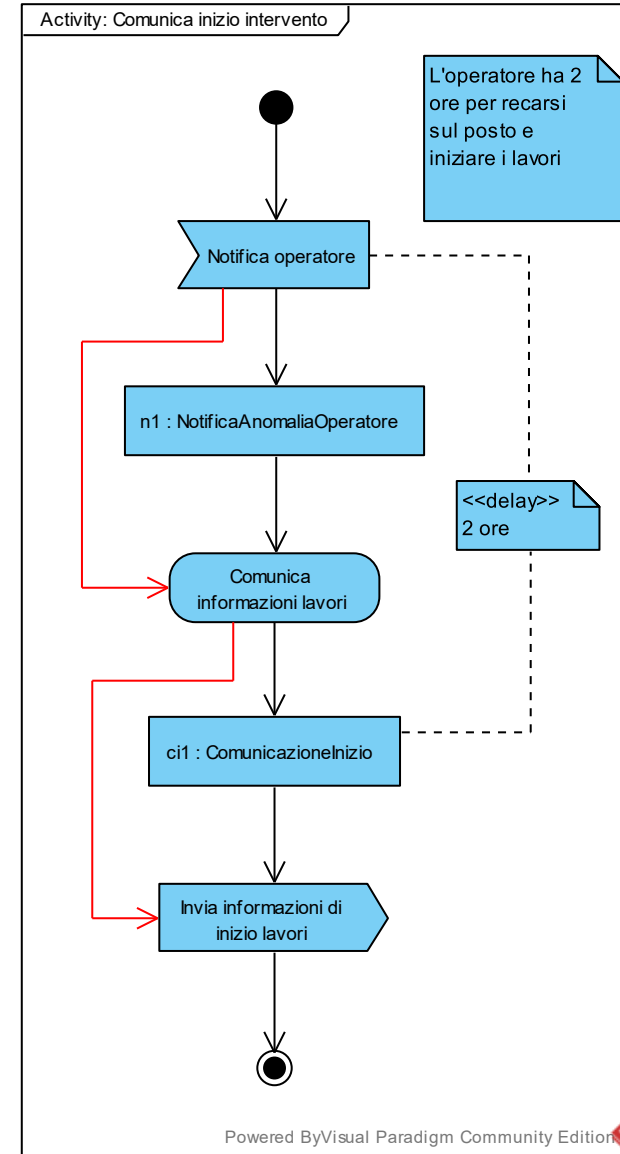
Identifica operatore + Notifica operatore

- Riceve un segnale dall'activity Verifica anomalia e l'oggetto AnomaliaDellaCentralina.
- Questo oggetto viene utilizzato per identificare l'operatore corretto.
 - L'operatore è quello più vicino alla centralina, che ha le capacità tecniche adatte ed è disponibile.
- Identifica operatore cerca nel datastore Operatori un operatore adatto. Nel caso non ci fosse un operatore disponibile, attende 15 minuti e riprova.
- Dopo aver identificato l'operatore, l'oggetto viene passato ad una azione che genera un altro oggetto chiamato NotificaAnomaliaOperatore, che contiene le informazioni necessarie per ripristinare la centralina.
- Questo oggetto viene inviato tramite il segnale Notifica operatore all'activity Comunicazione inizio intervento.



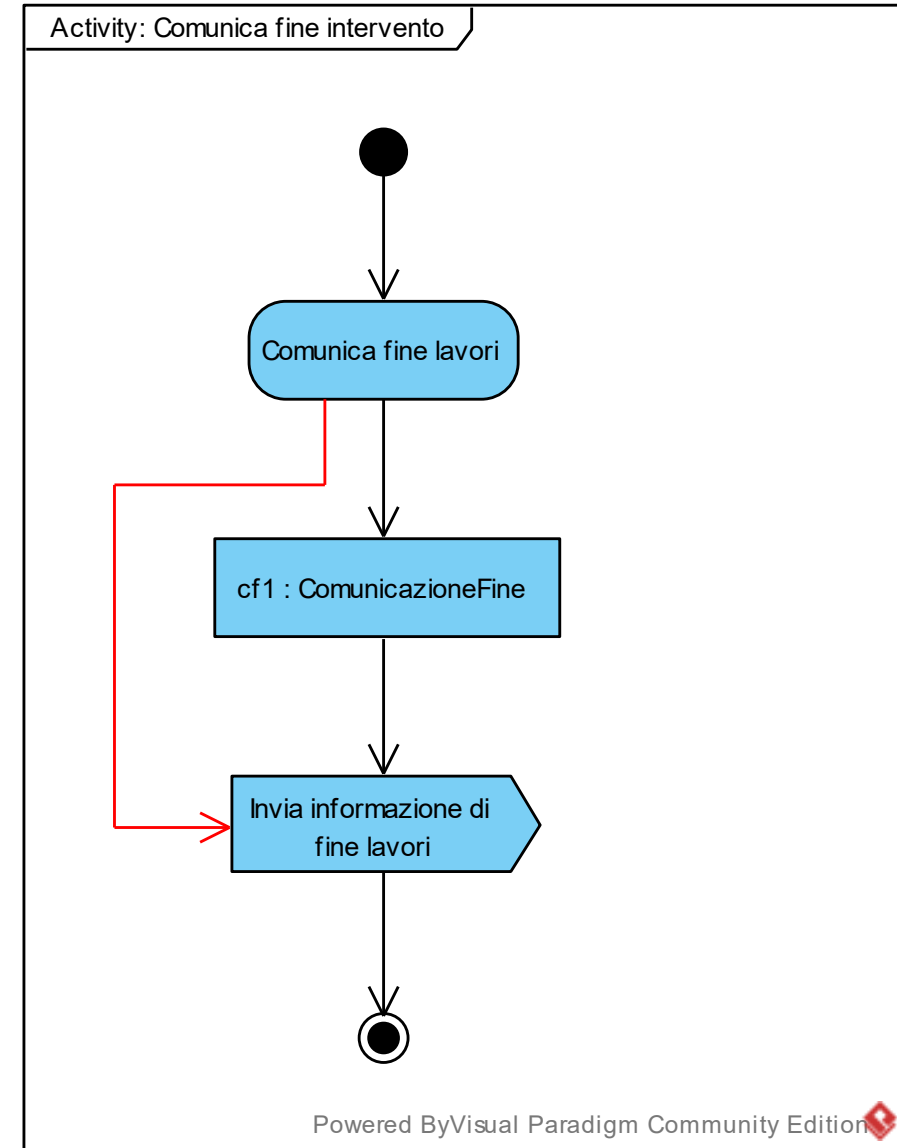
Comunica inizio intervento

- Riceve il segnale Notifica operatore e l'oggetto NotificaAnomaliaOperatore dall'activity Notifica operatore.
- L'operatore dopo la ricezione della notifica è stimato che ci metta in media due ore per recarsi alla centralina e comunicare l'inizio dei lavori.
- L'oggetto Comunicazione inizio, viene inviato al servizio tecnico centrale.



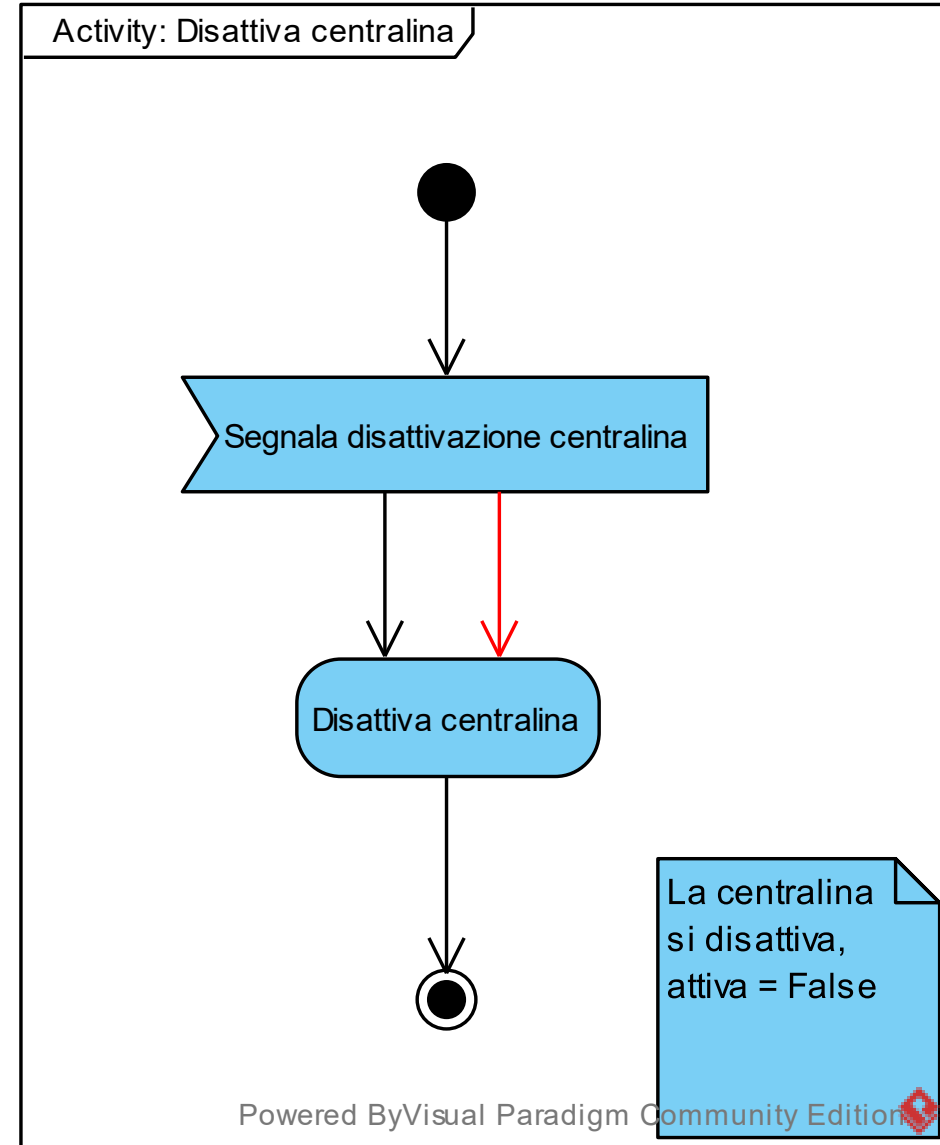
Comunica fine intervento

- Alla fine dei lavori, viene creato un oggetto Comunicazione fine, che viene inviato al servizio tecnico centrale.



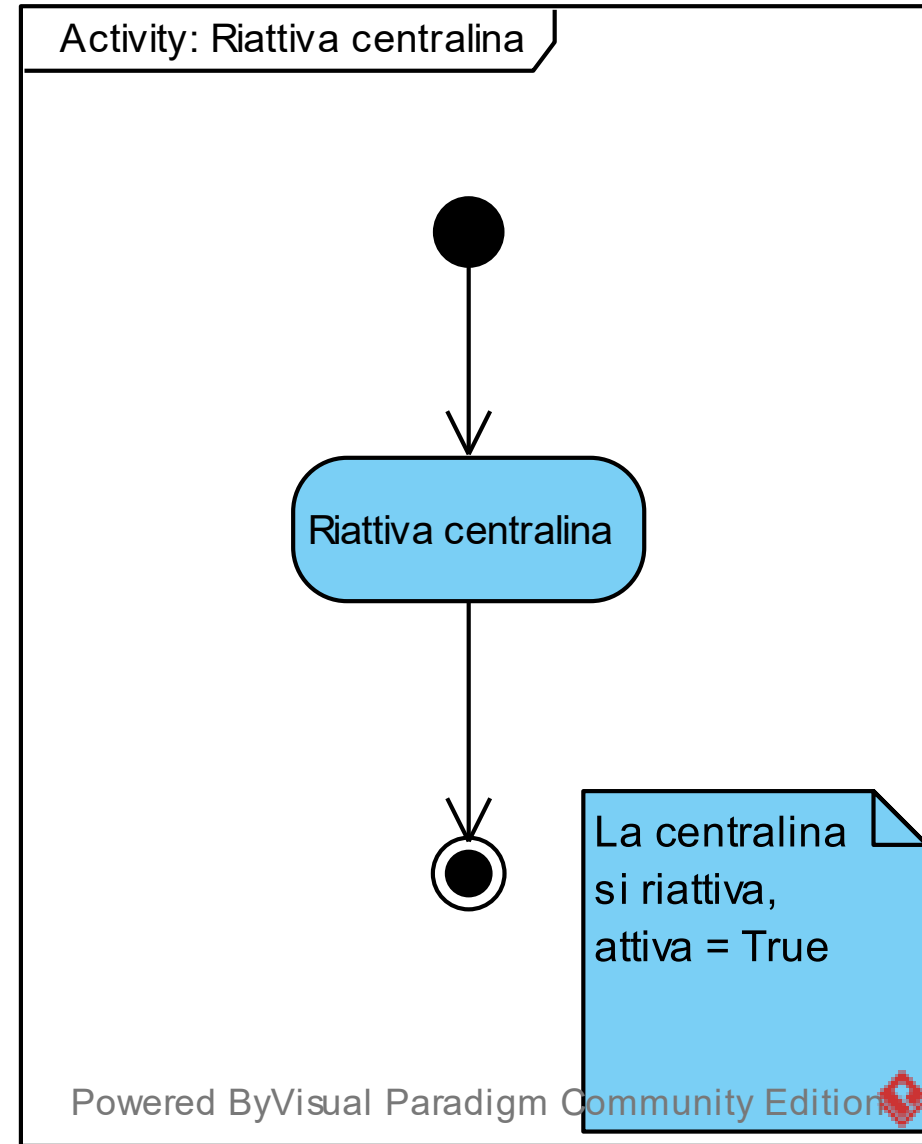
Disattiva centralina

- Riceve il segnale dall'activity Verifica anomalia.
- Disattiva la centralina a livello software, andando a cambiare la variabile "attiva" della Centralina da True a False.



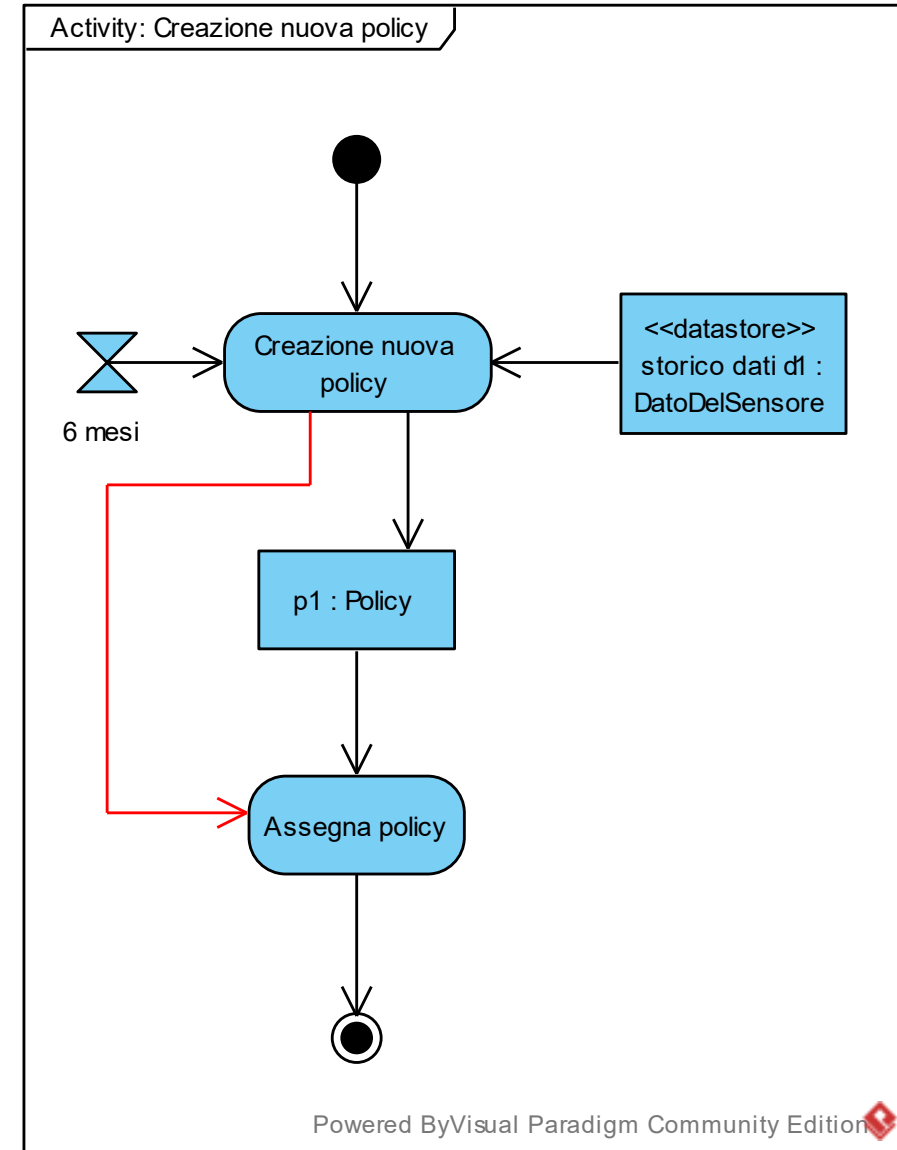
Riattiva centralina

- La centralina viene riattivata. La variabile “attiva” della Centralina, passa da False a True.



Creazione nuova policy

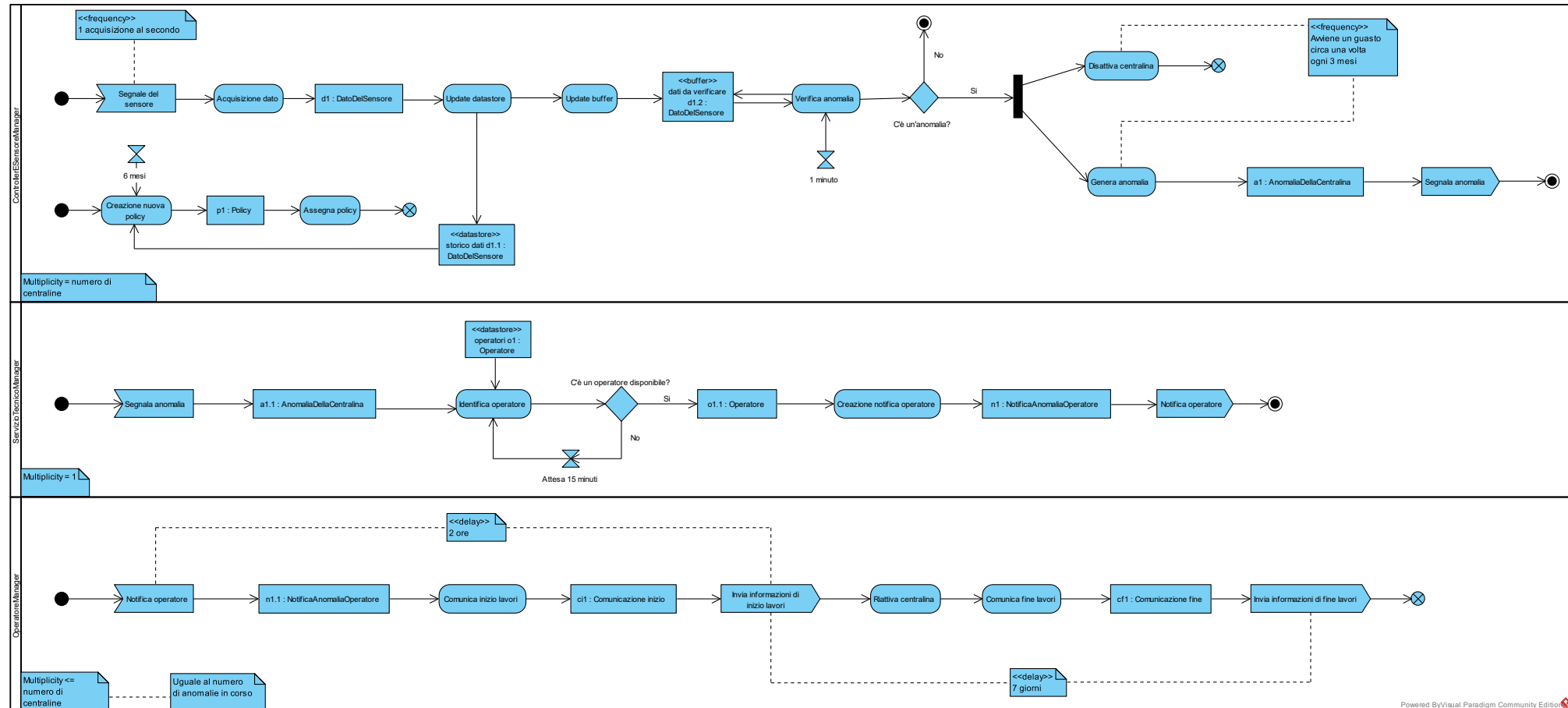
- Una volta ogni 6 mesi viene generata una nuova policy basandosi sui dati presenti nel datastore storico dati.
- Il datastore è associato ad ogni centralina, viene riempito in Acquisizione Dato.
- Dopo che è stato generato l'oggetto Policy, viene assegnato alla centralina.



3. Architettura Logica

- In questo capitolo vengono identificate le componenti logiche, dove ogni componente è un insieme di attività.
- Abbiamo identificato 3 footprint possibili e li abbiamo messi a confronto per capire quale fosse il migliore.

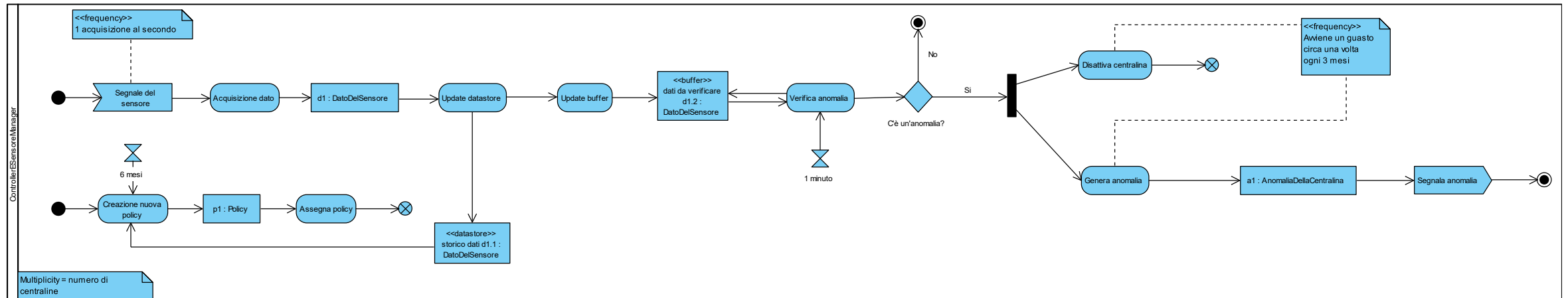
Footprint 1



Footprint 1

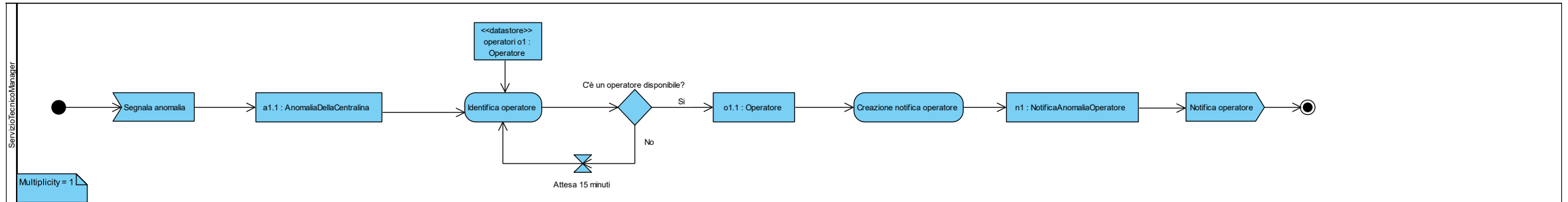
In questo footprint vengono identificate 3 componenti logiche:

- ControllerESensoreManager
- ServizioTecnicoManager
- OperatoreManager



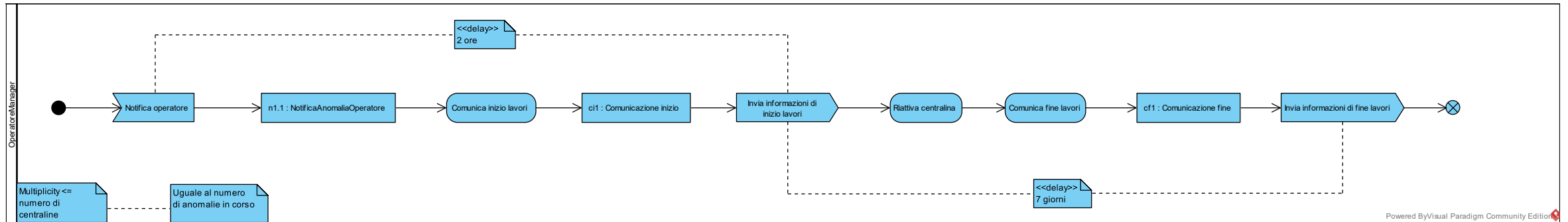
ControllerESensoreManager

- La prima componente logica riguarda il Sensore e il Controller.
- Comprende le activity Acquisizione dato, Verifica anomalia, Disattiva centralina e Creazione nuova policy.
- Una volta al secondo viene acquisito un dato, inserito nel datastore Storico dati e nel buffer Dati da verificare.
- Una volta al minuto viene controllato l'array di dati, preso dal buffer, e in seguito viene svuotato il buffer.
 - Nel caso si verificasse una anomalia, in parallelo:
 - Viene disattivata la centralina.
 - Viene generato un oggetto AnomaliaDellaCentralina e viene inviato il segnale dell'anomalia.
 - Nel caso non si verificasse una anomalia, l'activity termina.
- Ogni sei mesi Creazione nuova policy utilizza i dati del datastore per creare una nuova policy e assegnarla.



ServizioTecnicoManager

- La seconda componente logica riguarda il Servizio tecnico.
- Comprende le activity Identifica operatore + Notifica operatore.
- Riceve un oggetto AnomaliaDellaCentralina dalla componente logica ControllerESensoreManager.
- Utilizzando AnomaliaDellaCentralina identifica un operatore adatto a ripristinare la centralina.
 - Se trova un operatore adatto e disponibile, genera un oggetto Operatore e crea un oggetto NotificaAnomaliaOperatore, poi invia la notifica all'operatore stesso.
 - Se non trova un operatore disponibile attende 15 minuti e poi riprova ad identificarlo.

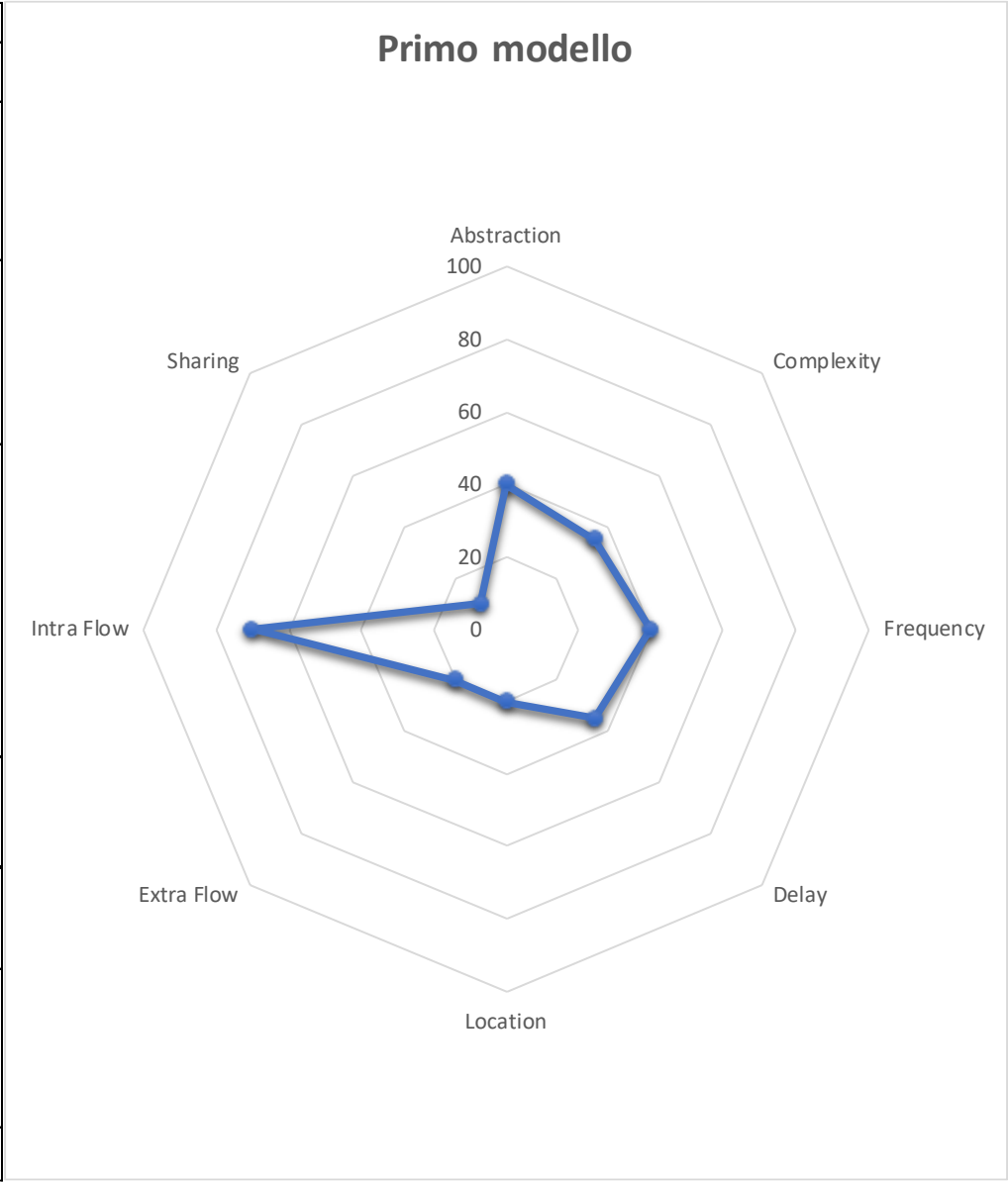


OperatoreManager

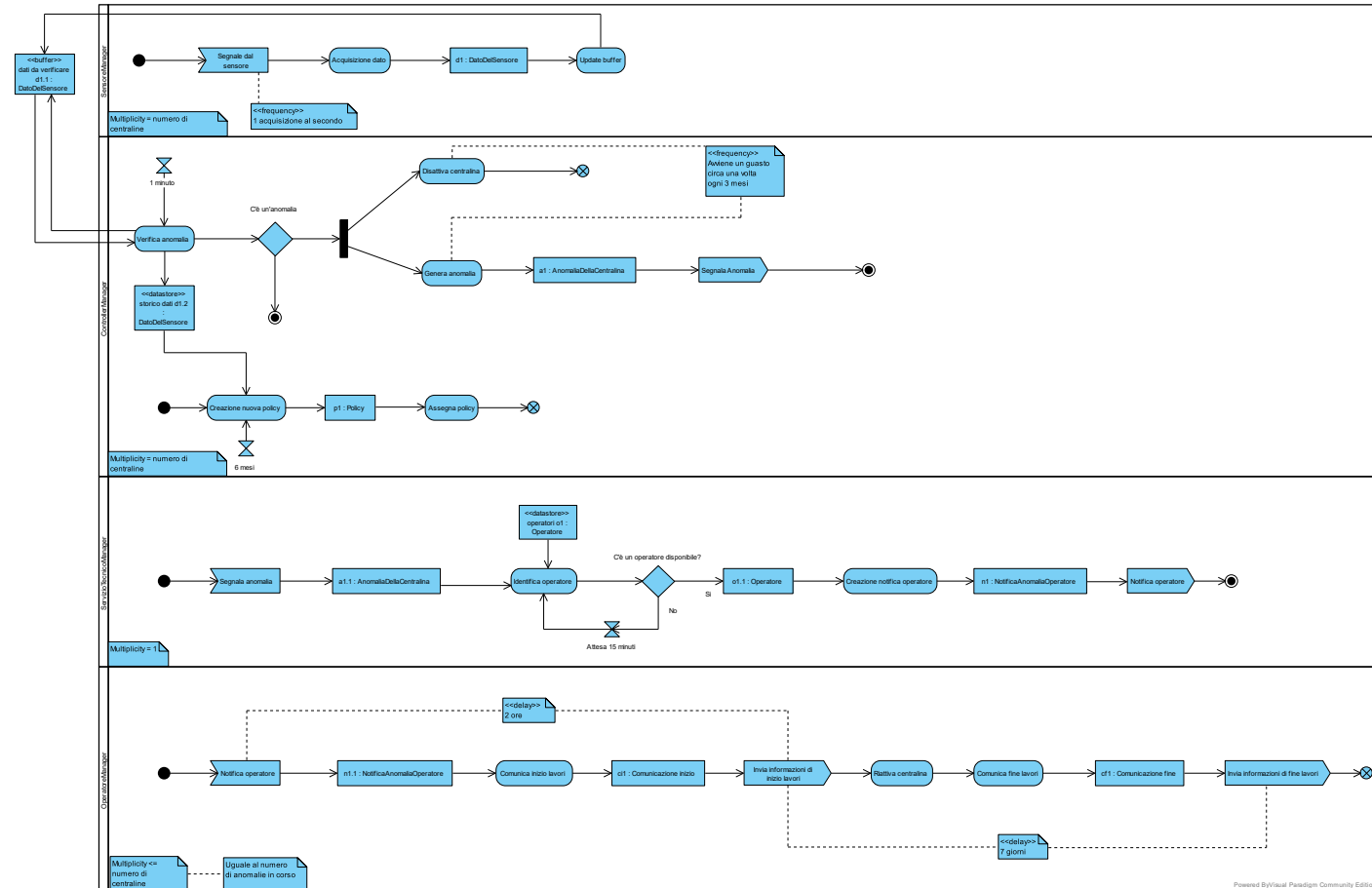
- La terza componente logica riguarda l'Operatore.
- Comprende le activity Comunica inizio lavori, Riattiva centralina e Comunica fine lavori.
- Riceve un oggetto NotificaAnomaliaOperatore dalla componente logica ServizioTecnicoManager.
- L'operatore ci mette in media due ore per recarsi sul posto di lavoro e comunicare l'inizio dei lavori.
- In media in 7 giorni, la centralina viene riattivata e l'operatore comunica la fine dei lavori.

Tabella Footprint 1

Metrica	Valore	Valore numerico	Motivazione
Abstraction	Medium	40	Perché ogni componente utilizza 3 dati.
Complexity	Medium/Low	35	Controller e sensore: spread medio, sono presenti azioni di complessità diverse. Servizio tecnico: spread basso. Operatore: spread basso.
Frequency	Medium	40	Controller e sensore: spread medio/alto (1 secondo, 1 minuti, 6 mesi). Servizio tecnico, Operatore: hanno la stessa frequenza, sono attività che si attivano in seguito ad una segnalazione.
Delay	Medium	35	Controller e sensore: spread medio/alto, acquisizione dato e verifica anomalia sono immediati, genera anomalia e disattiva centralina sono veloci, mentre creazione nuova policy richiede più tempo. Servizio tecnico: spread basso, sia identifica operatore che notifica operatore richiedono poco tempo. Operatore: spread medio, il tempo per iniziare i lavori è di qualche ora, mentre il tempo per terminare i lavori è di qualche giorno.
Location	Low	20	Ogni istanza comunica con attori collocati
Extra Flow	Low	20	Ogni componente interagisce con il proprio attore che è multi istanziato. C'è un attore per ogni istanza della componente.
Intra Flow	High	70	Tutte le istanze di ControllerESensoreManager comunicano con l'istanza di ServizioTecnicoManager. Tutte le istanze di OperatoreManager comunicano con l'istanza di ServizioTecnicoManager
Sharing	Low	10	I datastore sono legati alla singola istanza di una centralina



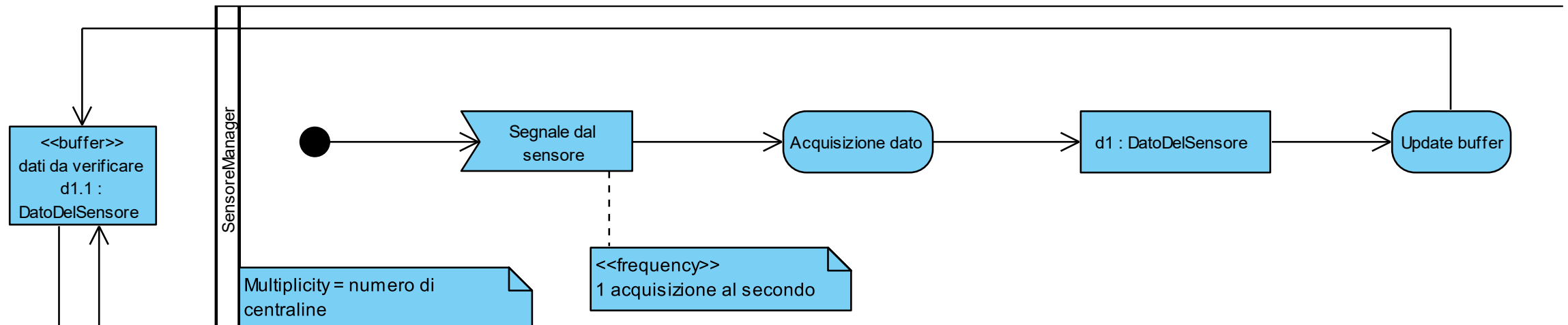
Footprint 2



Footprint 2

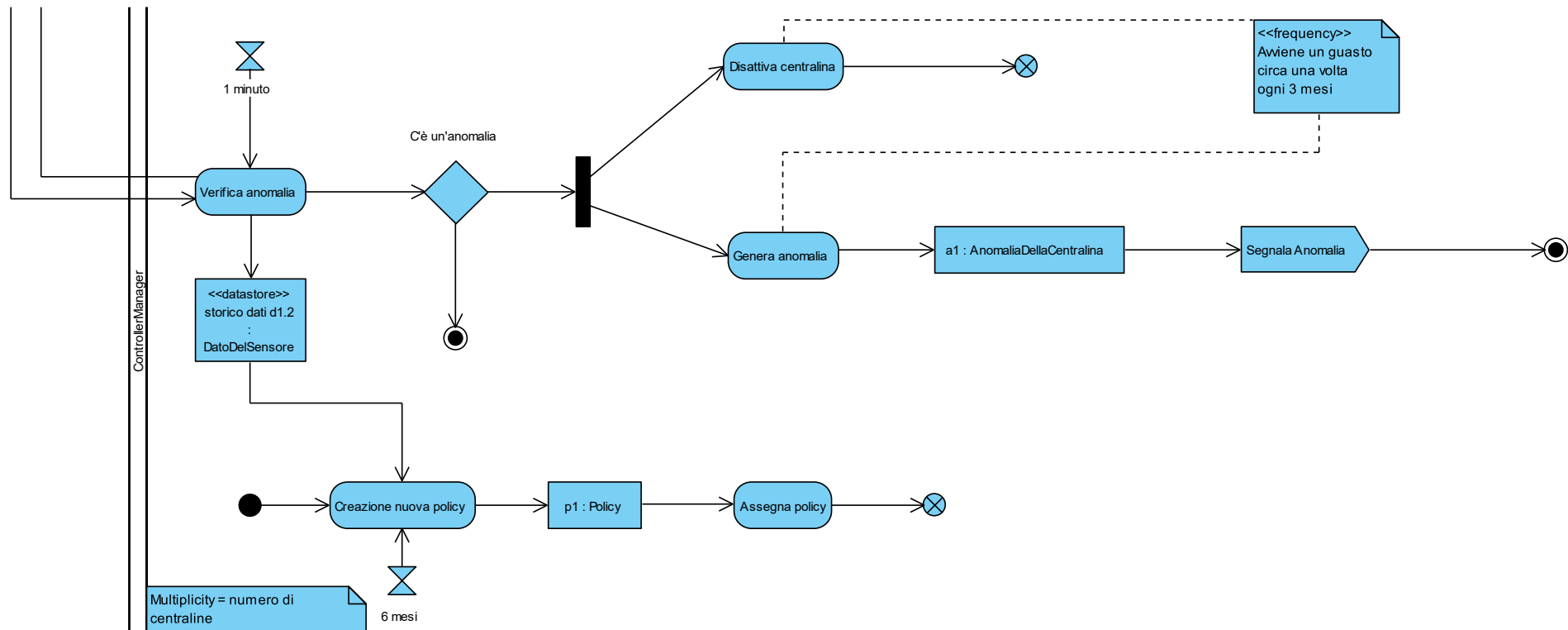
In questo footprint vengono identificate 3 componenti logiche:

- SensoreManager
- ControllerManager
- ServizioTecnicoManager
- OperatoreManager



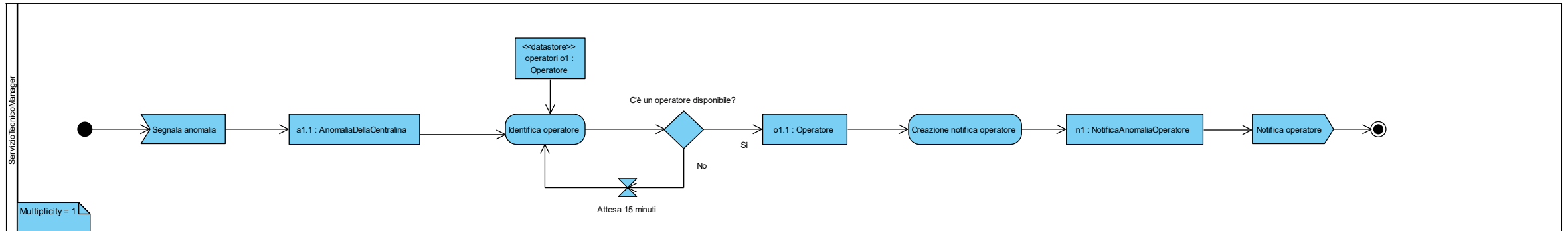
SensoreManager

- La prima componente logica riguarda il Sensore.
- Comprende l'activity Acquisizione dato.
- Una volta al secondo viene acquisito un dato e inserito nel buffer Dati da verificare.



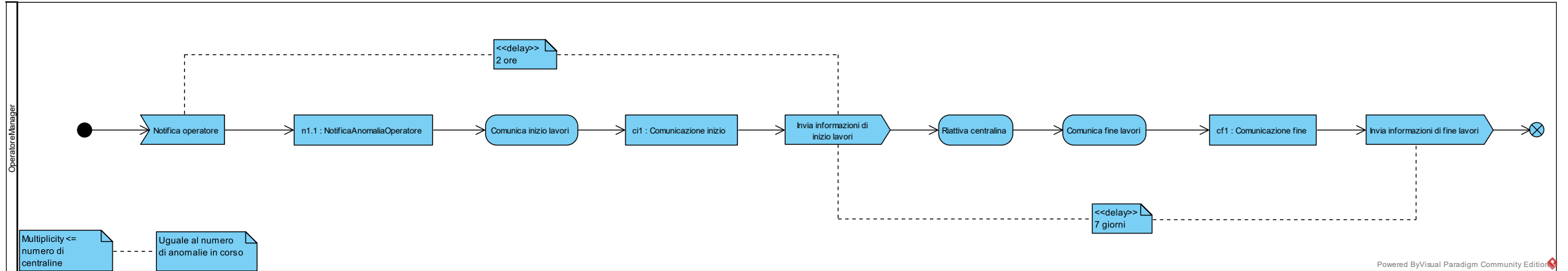
ControllerManager

- La seconda componente logica riguarda il Controller.
- Comprende le activity Verifica anomalia, Disattiva centralina e Creazione nuova policy.
- Una volta al minuto viene controllato un array di dati, preso dal buffer, svuotato il buffer e inserito l'array di dati nel datastore Storico dati.
 - Nel caso si verificasse una anomalia, in parallelo:
 - Viene disattivata la centralina.
 - Viene generato un oggetto AnomaliaDellaCentralina e viene inviato il segnale.
 - Nel caso non si verificasse una anomalia, l'activity termina.
- Ogni sei mesi Creazione nuova policy utilizza i dati nel datastore per creare una nuova policy e assegnarla.



ServizioTecnicoManager

- La terza componente logica riguarda il Servizio tecnico.
- Comprende le activity Identifica operatore + Notifica operatore.
- Riceve un oggetto AnomaliaDellaCentralina dalla componente logica ControllerESensoreManager.
- Utilizzando AnomaliaDellaCentralina identifica un operatore adatto a ripristinare la centralina.
 - Se trova un operatore adatto e disponibile, genera un oggetto Operatore e crea un oggetto NotificaAnomaliaOperatore, poi invia la notifica all'operatore stesso.
 - Se non trova un operatore disponibile attende 15 minuti e poi riprova ad identificarlo.

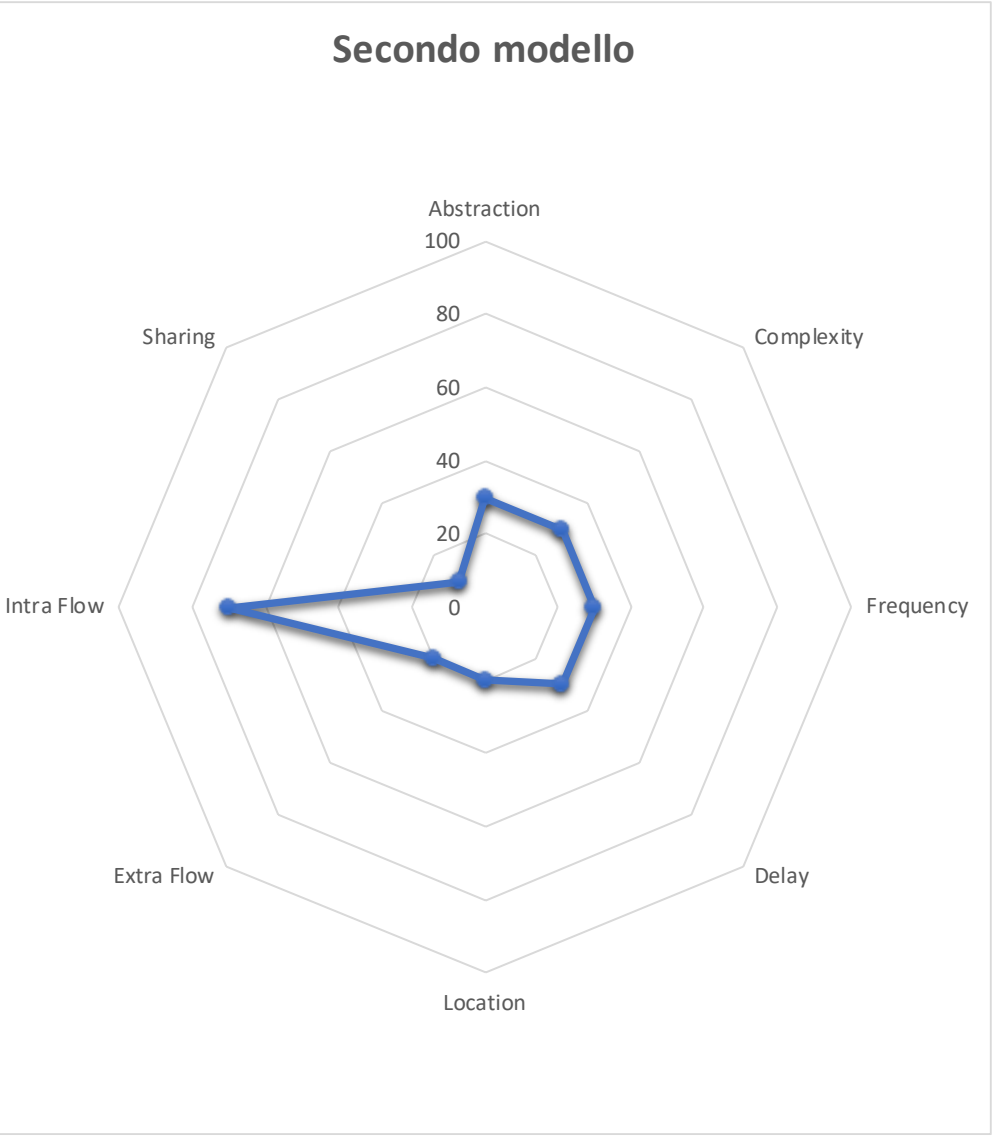


OperatoreManager

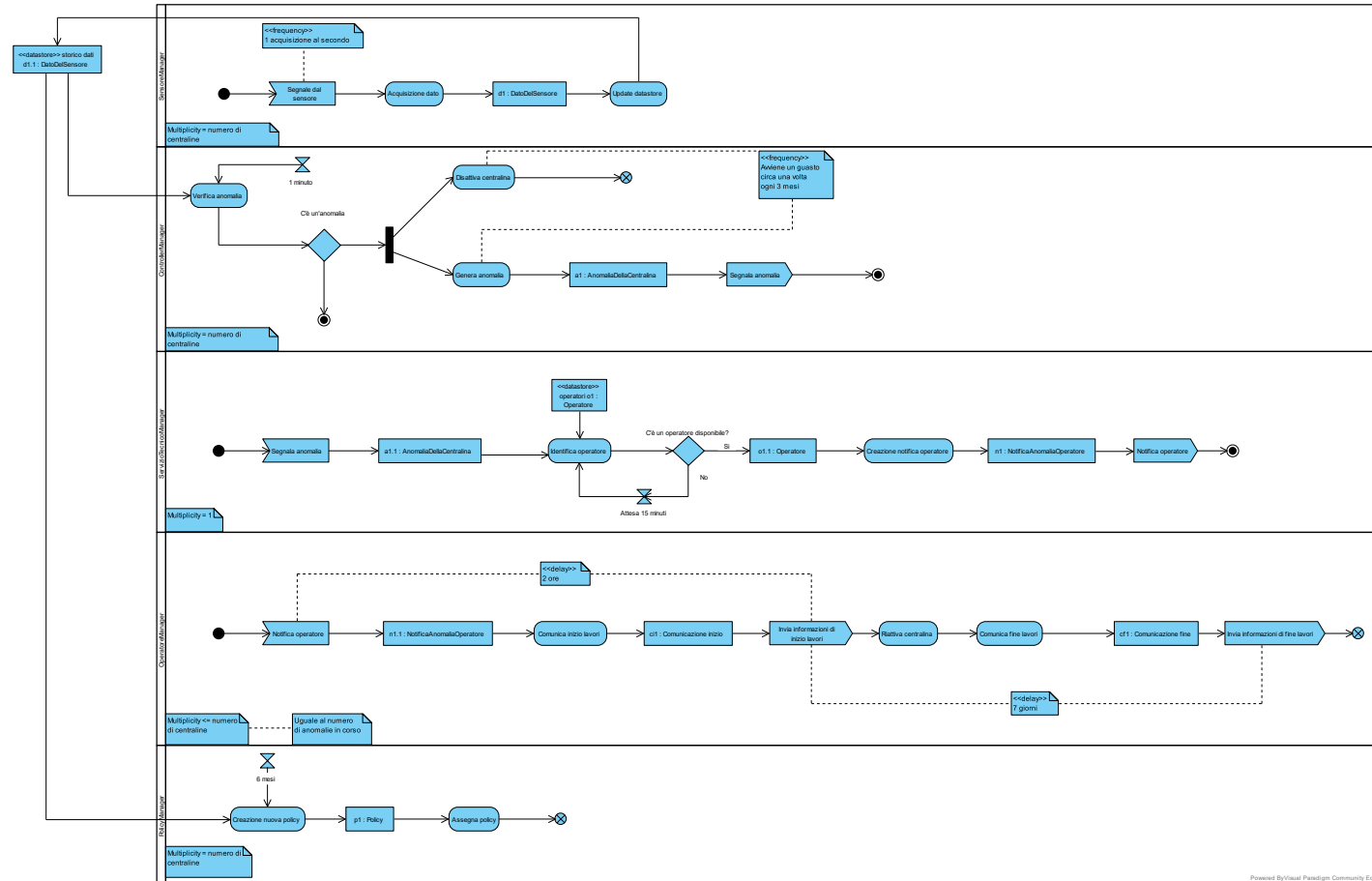
- La quarta componente logica riguarda l'Operatore.
- Comprende le activity Comunica inizio lavori, Riattiva centralina e Comunica fine lavori.
- Riceve un oggetto NotificaAnomaliaOperatore dalla componente logica ServizioTecnicoManager.
- L'operatore ci mette in media due ore per recarsi sul posto di lavoro e comunicare l'inizio dei lavori.
- In media in 7 giorni, la centralina viene riattivata e l'operatore comunica la fine dei lavori.

Tabella Footprint 2

Metrica	Valore	Valore numerico	Motivazione
Abstraction	Medium/Low	30	Perché la prima componente utilizza 1 dato, la seconda 2 dati mentre le altre due 3 dati.
Complexity	Medium/Low	30	Sensore: spread basso. Controller: spread medio/basso, sono presenti azioni di complessità diverse. Servizio tecnico: spread basso. Operatore: spread basso.
Frequency	Medium/Low	30	Sensore: ha una sola frequenza. Controller: spread medio (1 minuti, 6 mesi). Servizio tecnico, Operatore: hanno la stessa frequenza, sono attività che si attivano in seguito ad una segnalazione.
Delay	Medium/Low	30	Sensore: spread basso. Controller: spread medio, verifica anomalia è immediato, genera anomalia e disattiva centralina sono veloci; creazione nuova policy richiedono più tempo. Servizio tecnico: spread basso, sia identifica operatore che notifica operatore richiedono poco tempo. Operatore: spread medio, il tempo per iniziare i lavori è di qualche ora, mentre il tempo per terminare i lavori è di qualche giorno.
Location	Low	20	Ogni istanza comunica con attori collocati
Extra Flow	Low	20	Ogni componente interagisce con il proprio attore che è multi istanziato. C'è un attore per ogni istanza della componente.
Intra Flow	High	70	Tutte le istanze di ControllerESensoreManager comunicano con l'istanza di ServizioTecnicoManager. Tutte le istanze di OperatoreManager comunicano con l'istanza di ServizioTecnicoManager
Sharing	Low	10	I datastore sono legati alla singola istanza di una centralina



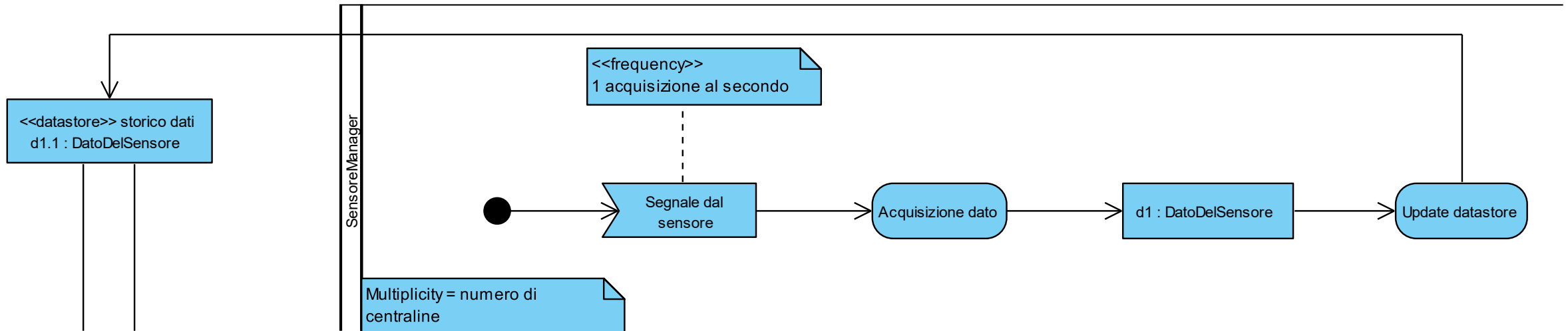
Footprint 3



Footprint 3

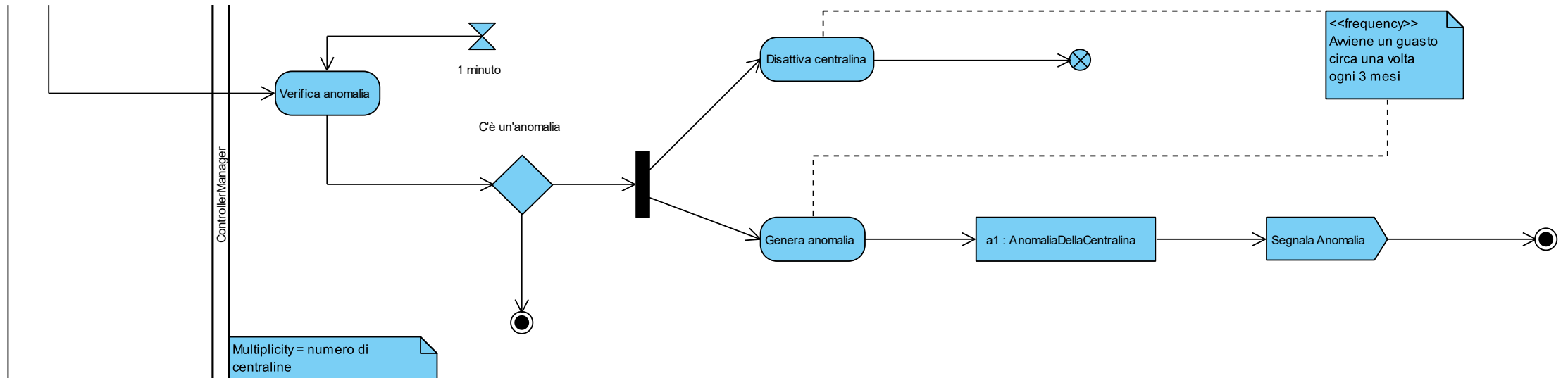
In questo footprint vengono identificate 5 componenti logiche:

- SensoreManager
- ControllerManager
- ServizioTecnicoManager
- OperatoreManager
- PolicyManager



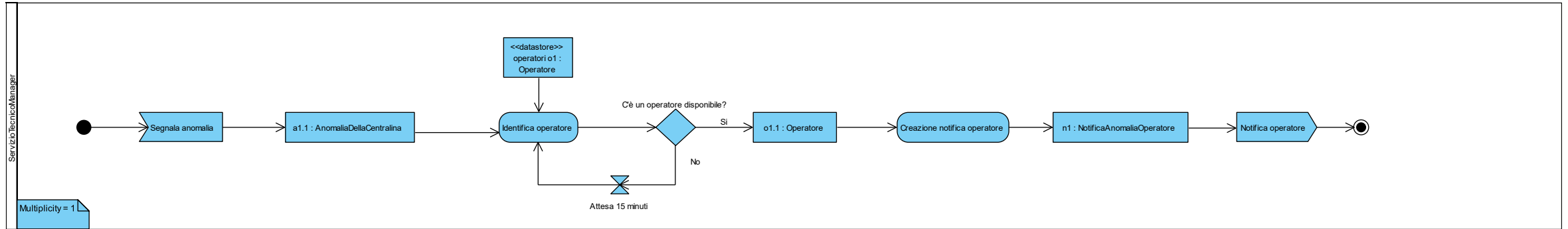
SensoreManager

- La prima componente logica riguarda il Sensore.
- Comprende l'activity Acquisizione dato.
- Una volta al secondo viene acquisito un dato e inserito nel datastore Storico dati.



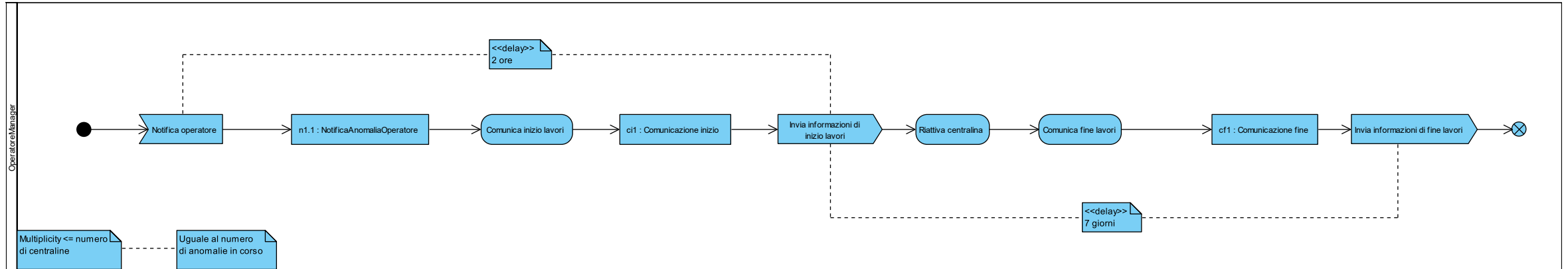
ControllerManager

- La seconda componente logica riguarda il Controller.
- Comprende le activity **Verifica anomalia**, **Disattiva centralina**.
- Una volta al minuto viene controllato un array di dati, preso dal datastore **Storico dati**.
- Nel caso si verificasse una anomalia, in parallelo:
 - Viene disattivata la centralina.
 - Viene generato un oggetto **AnomaliaDellaCentralina** e viene inviato il segnale.
- Nel caso non si verificasse una anomalia, l'activity termina.



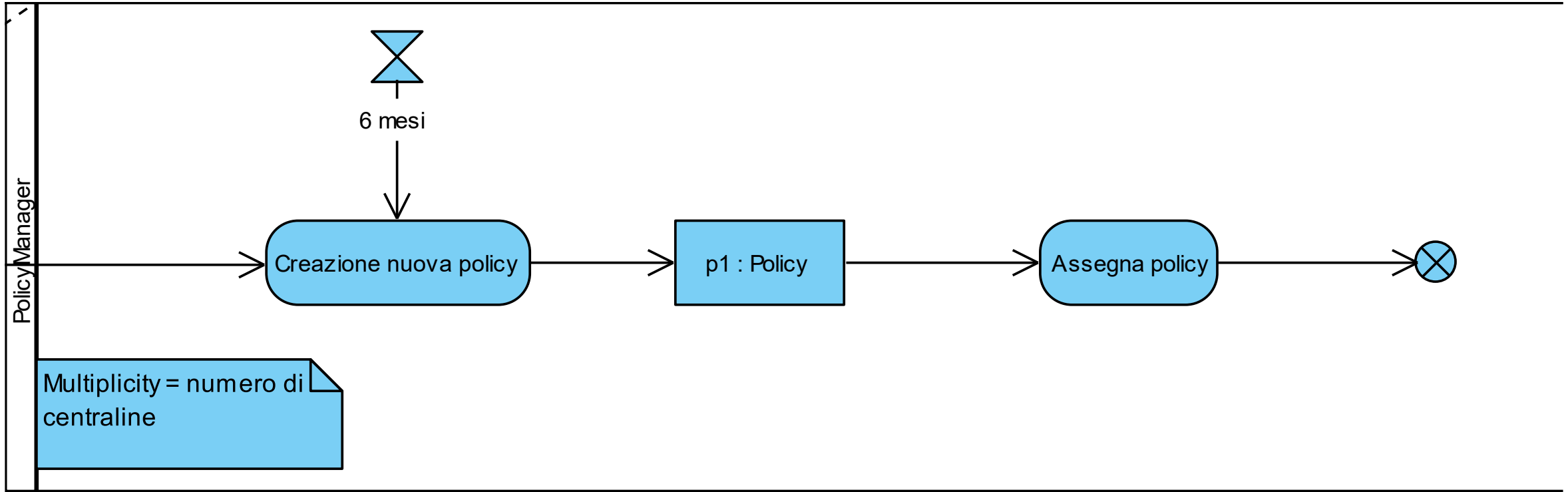
ServizioTecnicoManager

- La terza componente logica riguarda il Servizio tecnico.
- Comprende le activity Identifica operatore + Notifica operatore.
- Riceve un oggetto AnomaliaDellaCentralina dalla componente logica ControllerESensoreManager.
- Utilizzando AnomaliaDellaCentralina identifica un operatore adatto a ripristinare la centralina.
 - Se trova un operatore adatto e disponibile, genera un oggetto Operatore e crea un oggetto NotificaAnomaliaOperatore, poi invia la notifica all'operatore stesso.
 - Se non trova un operatore disponibile attende 15 minuti e poi riprova ad identificarlo.



OperatoreManager

- La quarta componente logica riguarda l'Operatore.
- Comprende le activity Comunica inizio lavori, Riattiva centralina e Comunica fine lavori.
- Riceve un oggetto NotificaAnomaliaOperatore dalla componente logica ServizioTecnicoManager.
- L'operatore ci mette in media due ore per recarsi sul posto di lavoro e comunicare l'inizio dei lavori.
- In media in 7 giorni, la centralina viene riattivata e l'operatore comunica la fine dei lavori.



PolicyManager

- La quinta componente riguarda il controller.
- Comprende l'activity Creazione nuova policy
- Ogni 6 mesi vengono utilizzati i dati del datastore Storico dati per generare una nuova policy e assegnarla.

Tabella Footprint 3

Metrica	Valore	Valore numerico	Motivazione
Abstraction	Low	25	Perché la prima, la seconda e l'ultima componente utilizzano 1 dato, mentre le altre due 3 dati.
Complexity	Low	20	Sensore: spread basso. Controller: spread basso. Servizio tecnico: spread basso. Operatore: spread basso. Policy: spread basso.
Frequency	Low	25	Sensore: ha una sola frequenza. Controller: ha una sola frequenza. Servizio tecnico, Operatore: hanno la stessa frequenza, sono attività che si attivano in seguito ad una segnalazione. Policy: ha una sola frequenza.
Delay	Low	25	Sensore: spread basso. Controller: spread medio/basso, verifica anomalia è immediato, genera anomalia e disattiva centralina sono veloci. Servizio tecnico: spread basso, sia identifica operatore che notifica operatore richiedono poco tempo. Operatore: spread medio, il tempo per iniziare i lavori è di qualche ora, mentre il tempo per terminare i lavori è di qualche giorno. Policy: spread basso.
Location	Low	20	Ogni istanza comunica con attori collocati
Extra Flow	Low	20	Ogni componente interagisce con il proprio attore che è multi istanziato. C'è un attore per ogni istanza della componente.
Intra Flow	High	70	Tutte le istanze di ControllerESensoreManager comunicano con l'istanza di ServizioTecnicoManager. Tutte le istanze di OperatoreManager comunicano con l'istanza di ServizioTecnicoManager
Sharing	Low	10	Tutte le istanze di OperatoreManager comunicano con l'istanza di ServizioTecnicoManager

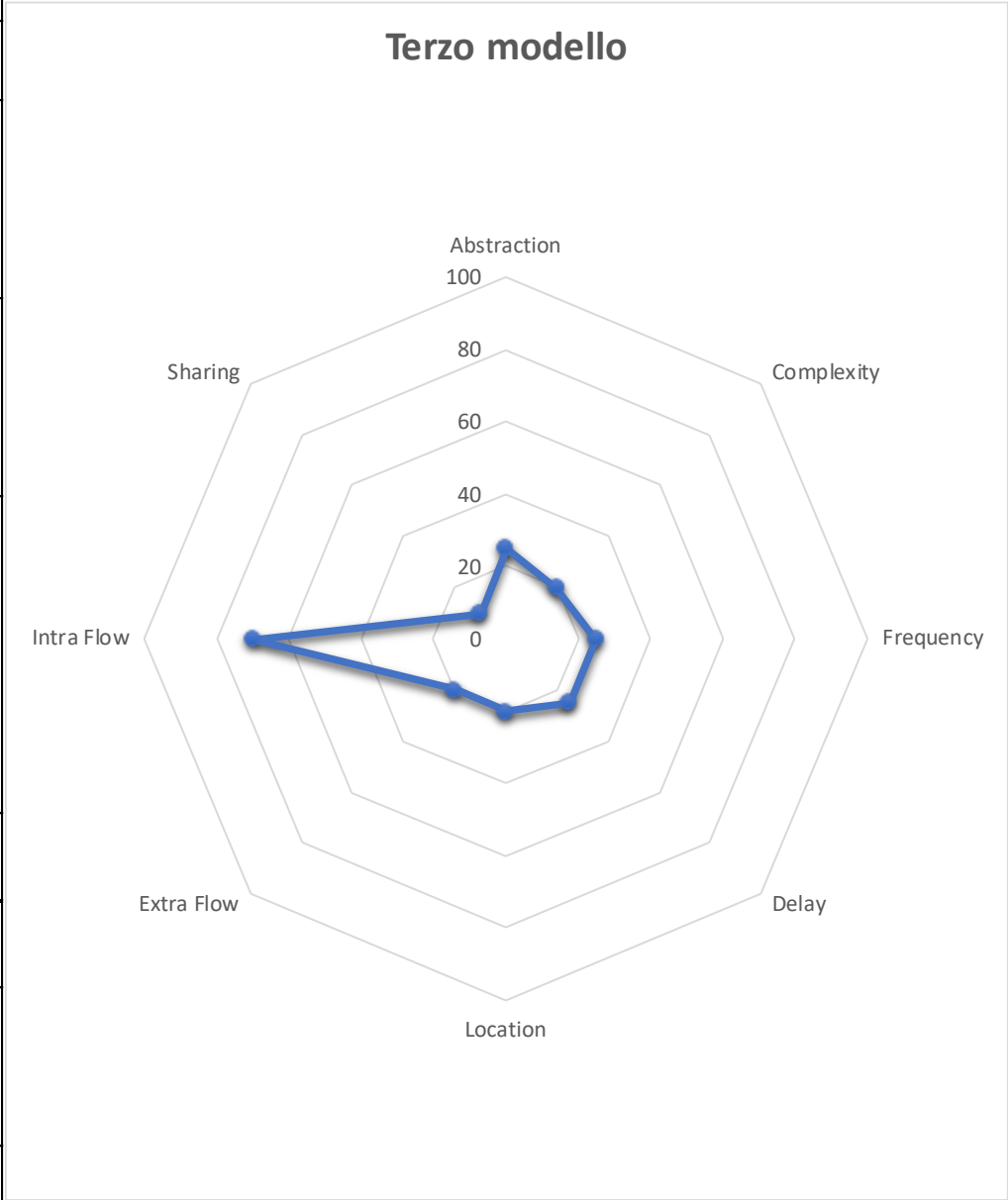


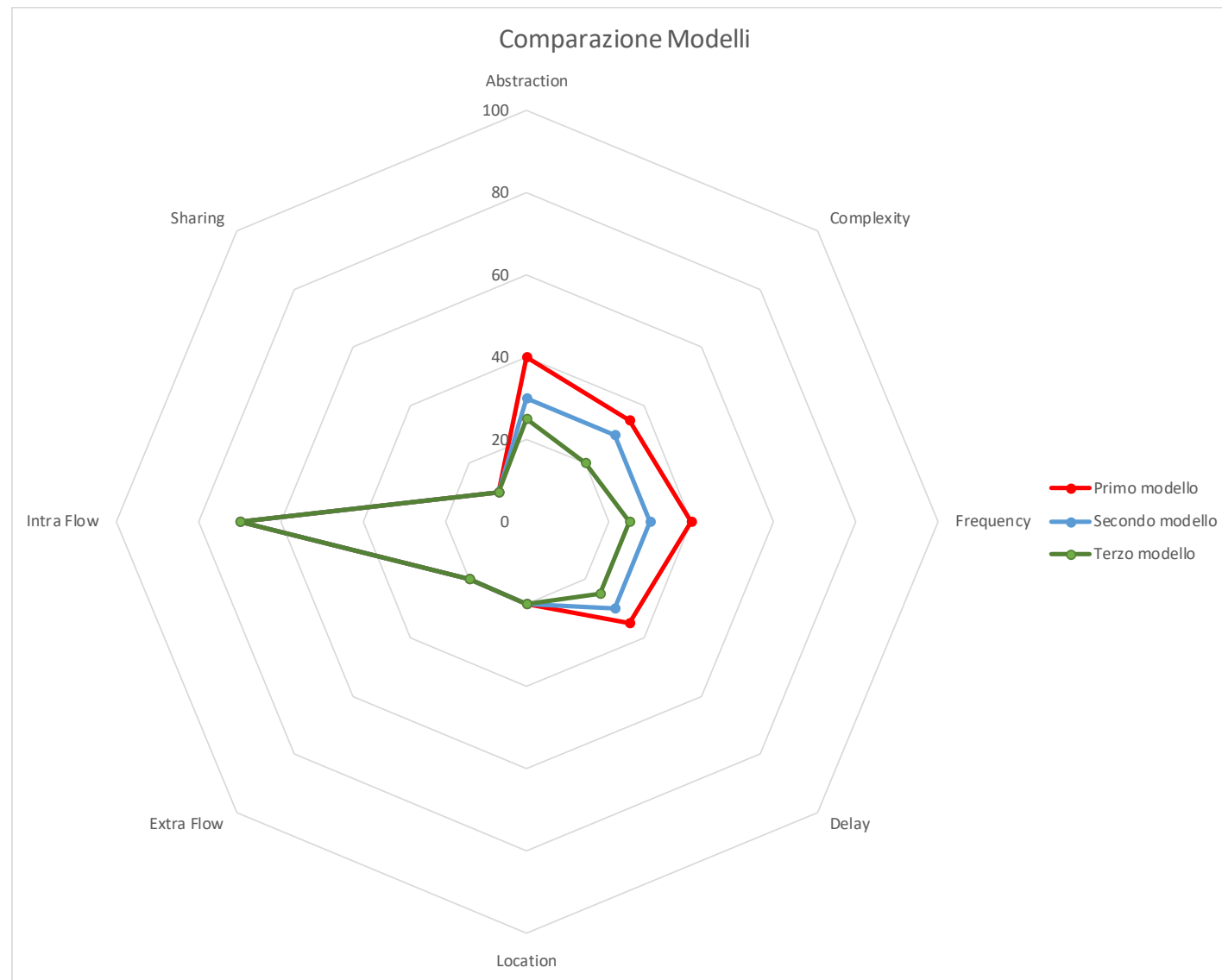
Grafico di comparazione tra i footprint

Il terzo modello è quello migliore, in quasi tutte le metriche ha valori più bassi.

I valori di Abstraction, Complexity, Frequency e Delay sono diminuiti, rispetto agli altri due modelli.

I valori di IntraFlow, ExtraFlow e Sharing rimangono uguali per tutti e tre i modelli.

Il terzo modello è stato quindi scelto come modello da utilizzare nell'architettura concreta.

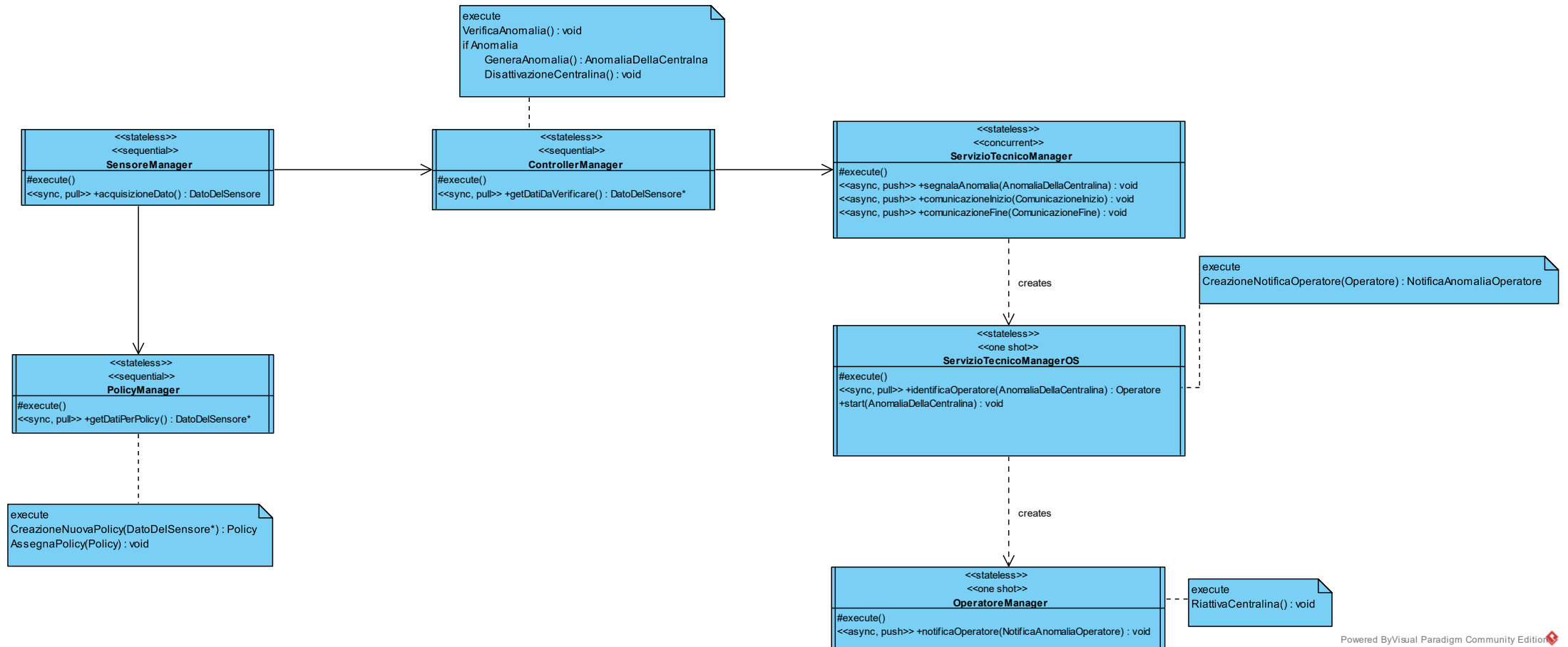


4. Architettura Concreta

In questo capitolo vengono viste le interazioni tra i componenti e per ogni componente viene visualizzato il design interno. Tutto questo tramite:

- Diagramma delle classi.
- Diagramma di sequenza.

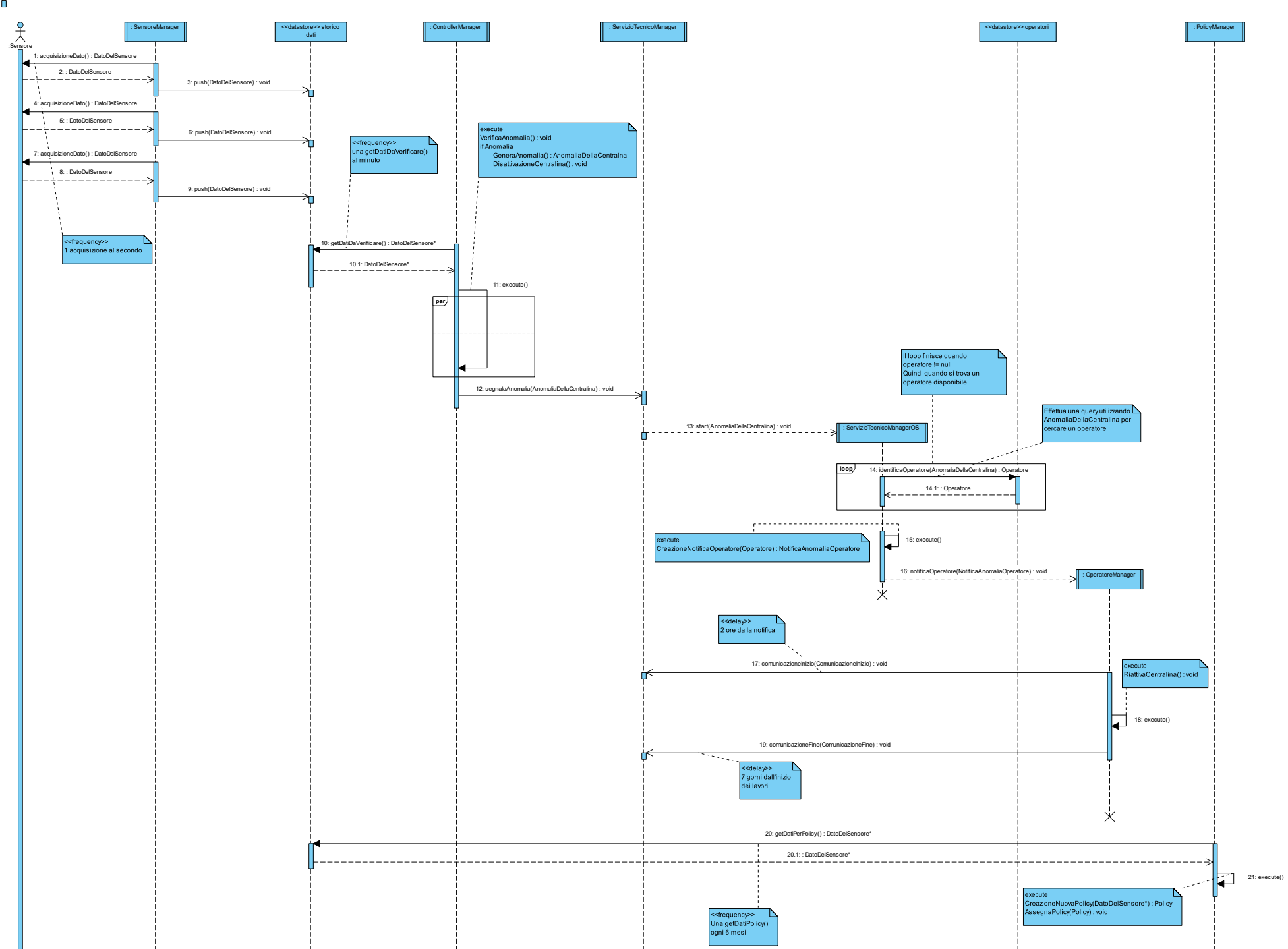
Diagramma delle classi



Descrizione componenti

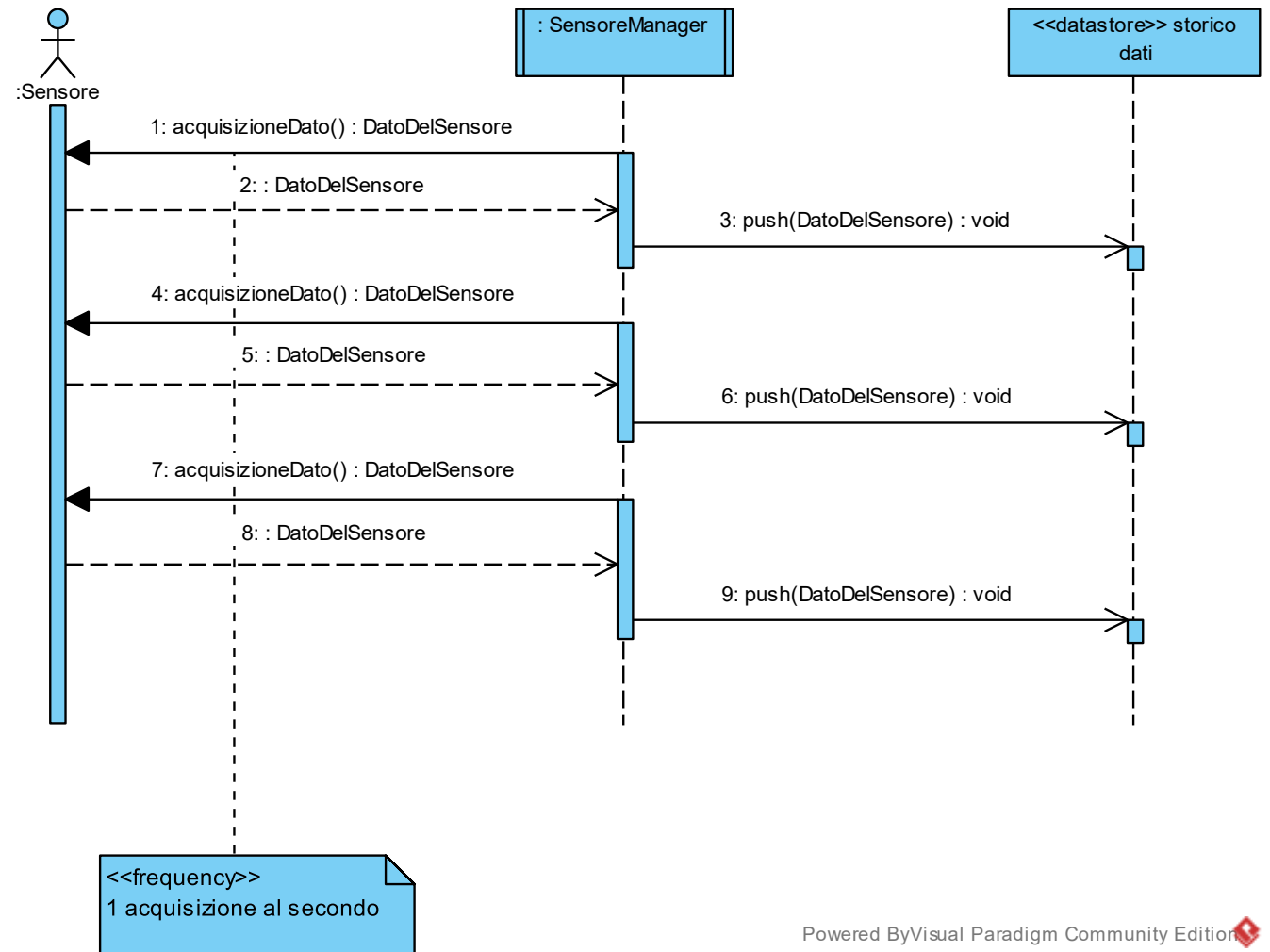
- Ogni componente è attiva, perché esegue le operazioni nel proprio thread.
- Ogni componente è stateless, perché non salva lo stato.
- ServizioTecnicoManager è concorrente. Dopo che gli arriva una anomalia, genera ServizioTecnicoManagerOS, una componente One Shot.
- ServizioTecnicoManagerOS identifica un operatore e poi crea OperatoreManager che è One Shot.
- Le altre componenti sono Sequenziali, perché eseguono operazioni in sequenza.

- Il diagramma di sequenza viene analizzato nelle prossime slide, componente per componente. Per ogni componente viene visualizzato anche quali sono le operazioni presenti nell'execute.
- In questo grafico (e nella prossima slide) invece viene visualizzato il flusso completo.



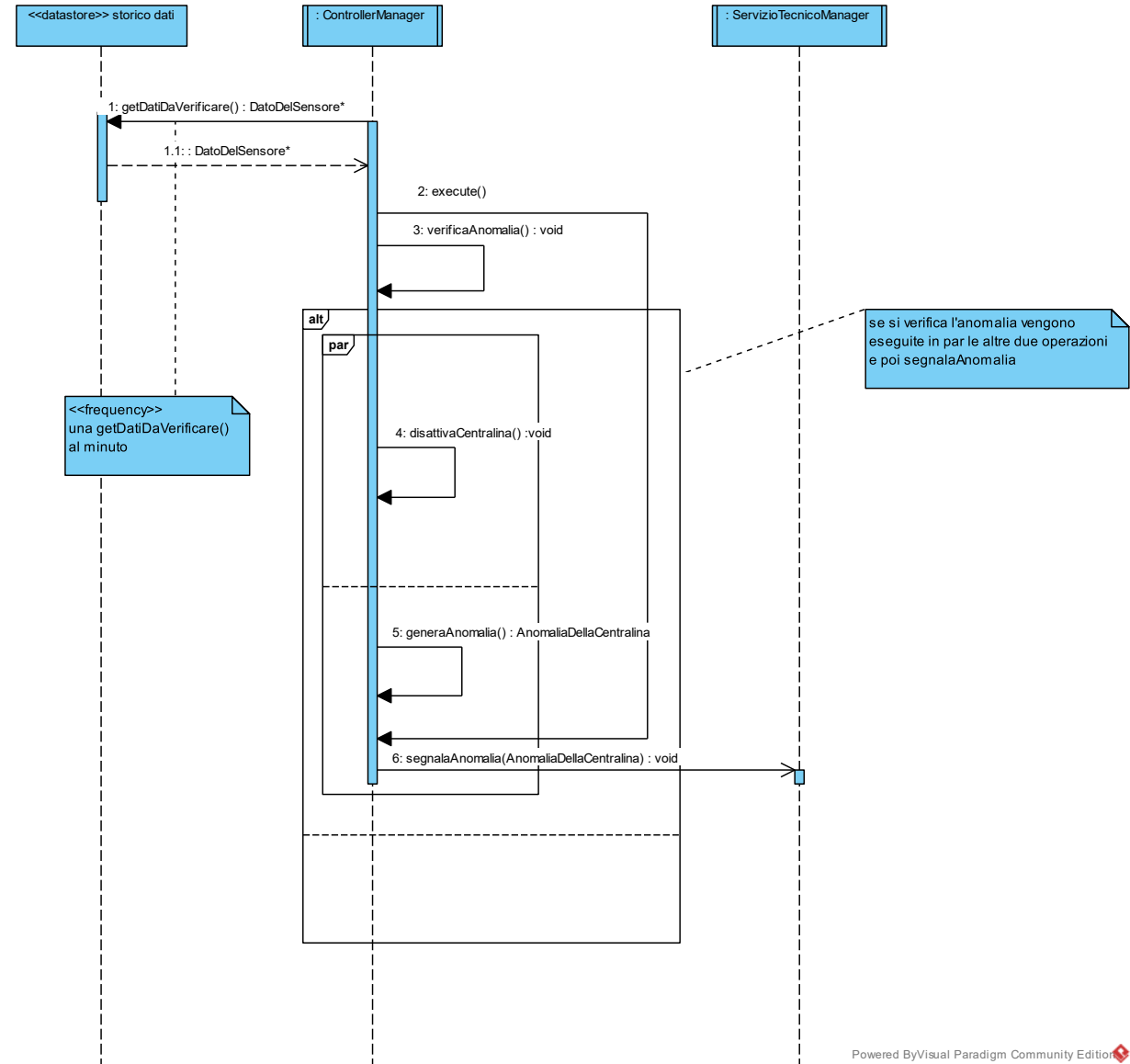
SensoreManager

- La componente SensoreManager è sequenziale.
- Tramite il metodo acquisizioneDato(), acquisisce il dato da un sensore fisico.
- Poi effettua una push per inserire il dato nel datastore Storico dati.



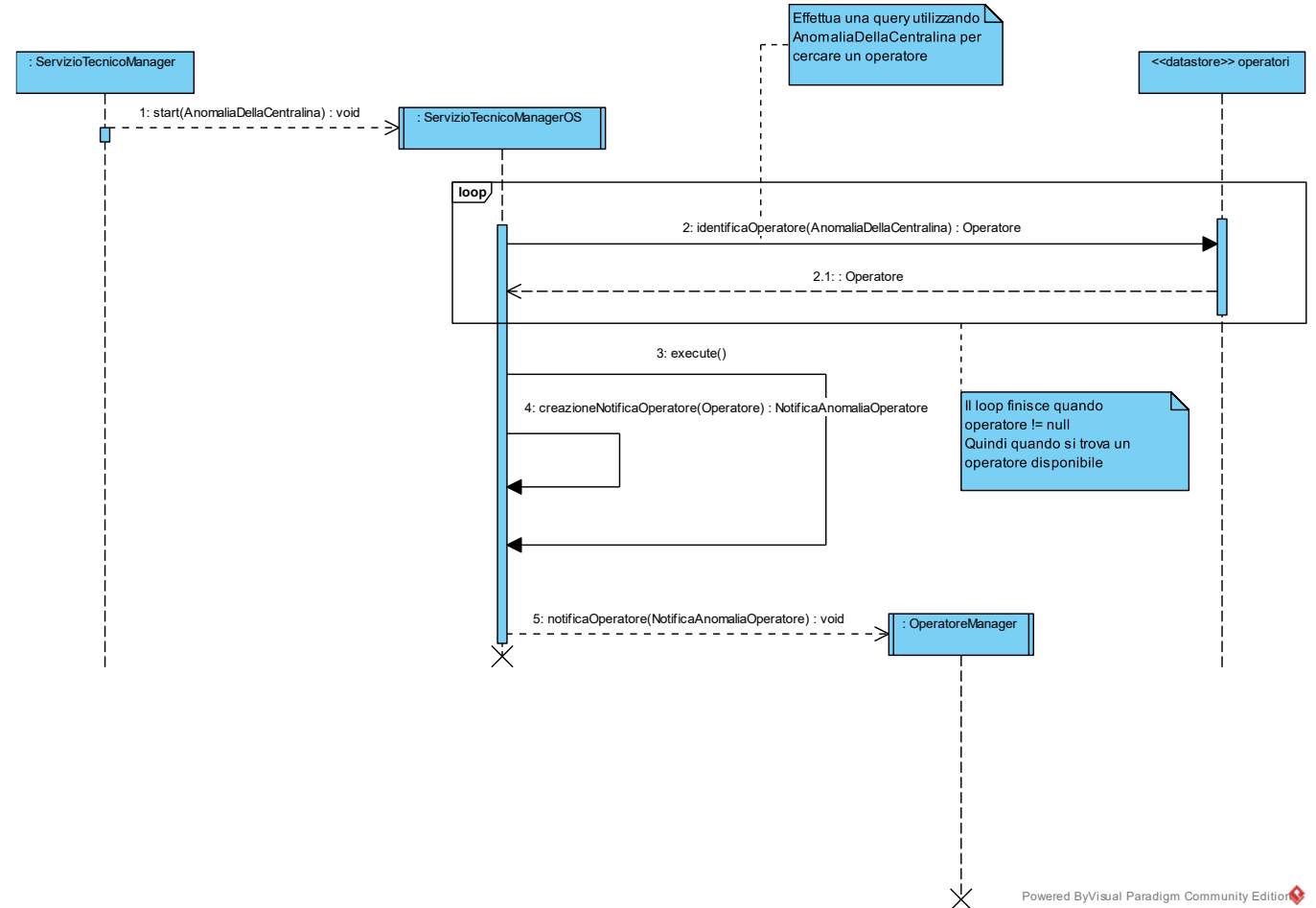
ControllerManager

- La componente ControllerManager è sequenziale.
- Ogni minuto effettua una pull per ottenere un array di dati da verificare.
- Effettua l'execute:
 - Verifica se c'è un'anomalia.
 - Se l'anomalia è presente, in parallelo disattiva la centralina e genera una anomalia.
 - Dopo aver generato l'anomalia, la invia al ServizioTecnicoManager.
 - Se non c'è un'anomalia non esegue nulla.



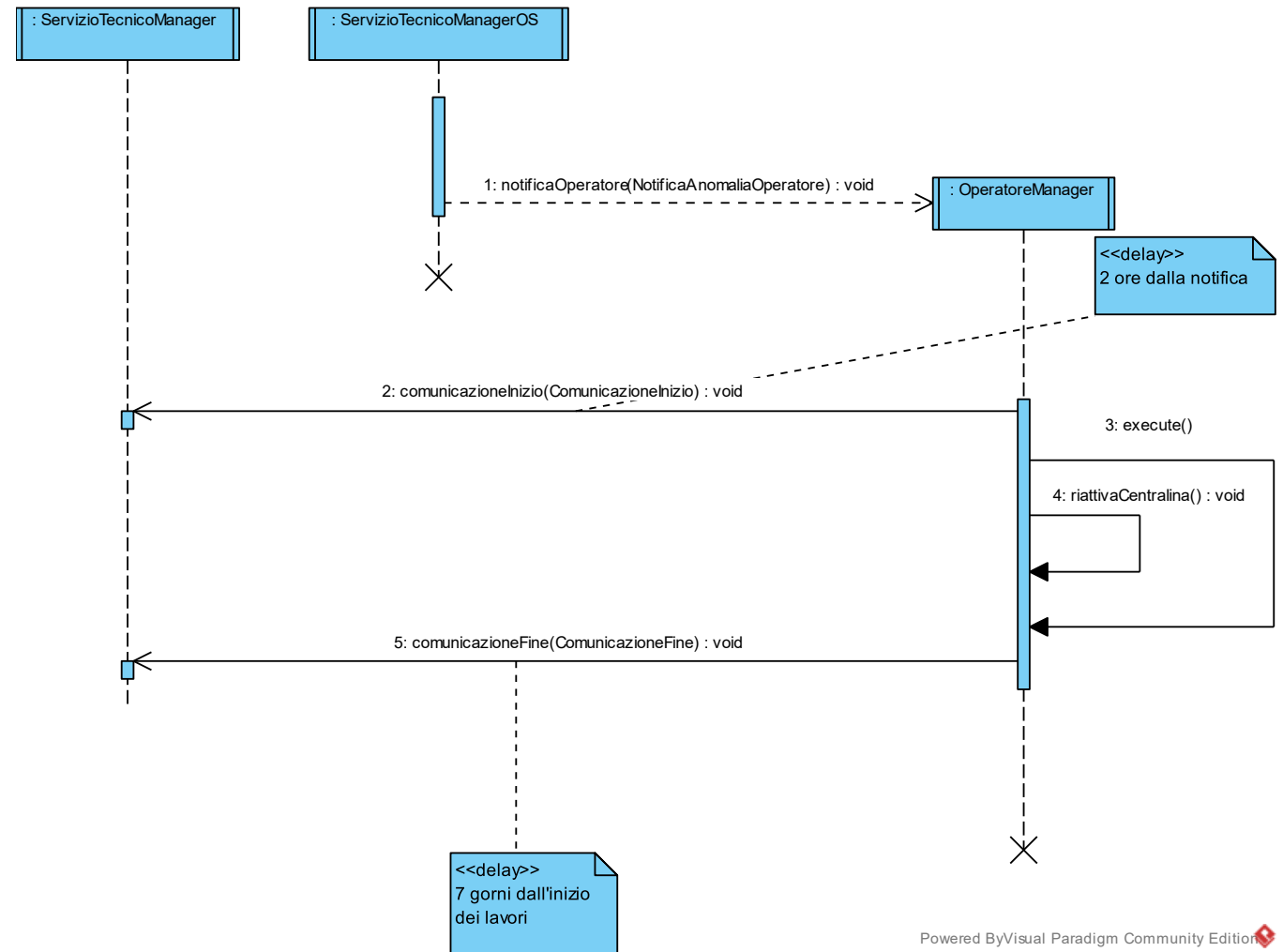
ServizioTecnicoManager

- La componente ServizioTecnicoManager è concorrente.
- Ogni volta che riceve una anomalia, genera ServizioTecnicoManagerOS, una componente One Shot.
- ServizioTecnicoManagerOS cerca un operatore adatto. Questa operazione è eseguita in un loop, questo finisce nel momento in cui l'operatore viene trovato.
- Poi viene eseguito l'execute:
 - Viene creata la notifica per l'operatore.
- Viene poi creata la componente One Shot OperatoreManager, tramite l'invio della notifica NotificaAnomaliaOperatore.



OperatoreManager

- La componente OperatoreManager è One Shot.
- Viene creata da ServizioTecnicoManagerOS tramite `notificaOperatore`.
- Inizialmente comunica le informazioni iniziali al ServizioTecnicoManager.
- Effettua l'`execute`:
 - Riattiva la centralina.
- Infine comunica le informazioni finali al ServizioTecnicoManager.



PolicyManager

- La componente PolicyManager è sequenziale.
- Ogni sei mesi effettua una pull per ottenere un array di dati da analizzare.
- Effettua l'execute:
 - Crea la nuova policy andando ad utilizzare i dati ottenuti dalla pull.
 - Assegna la policy alla centralina.

