Factory 4.0 Fabricated

Di seguito i documenti di progetto:

- Flusso informativo
- Requisiti industria 4.0

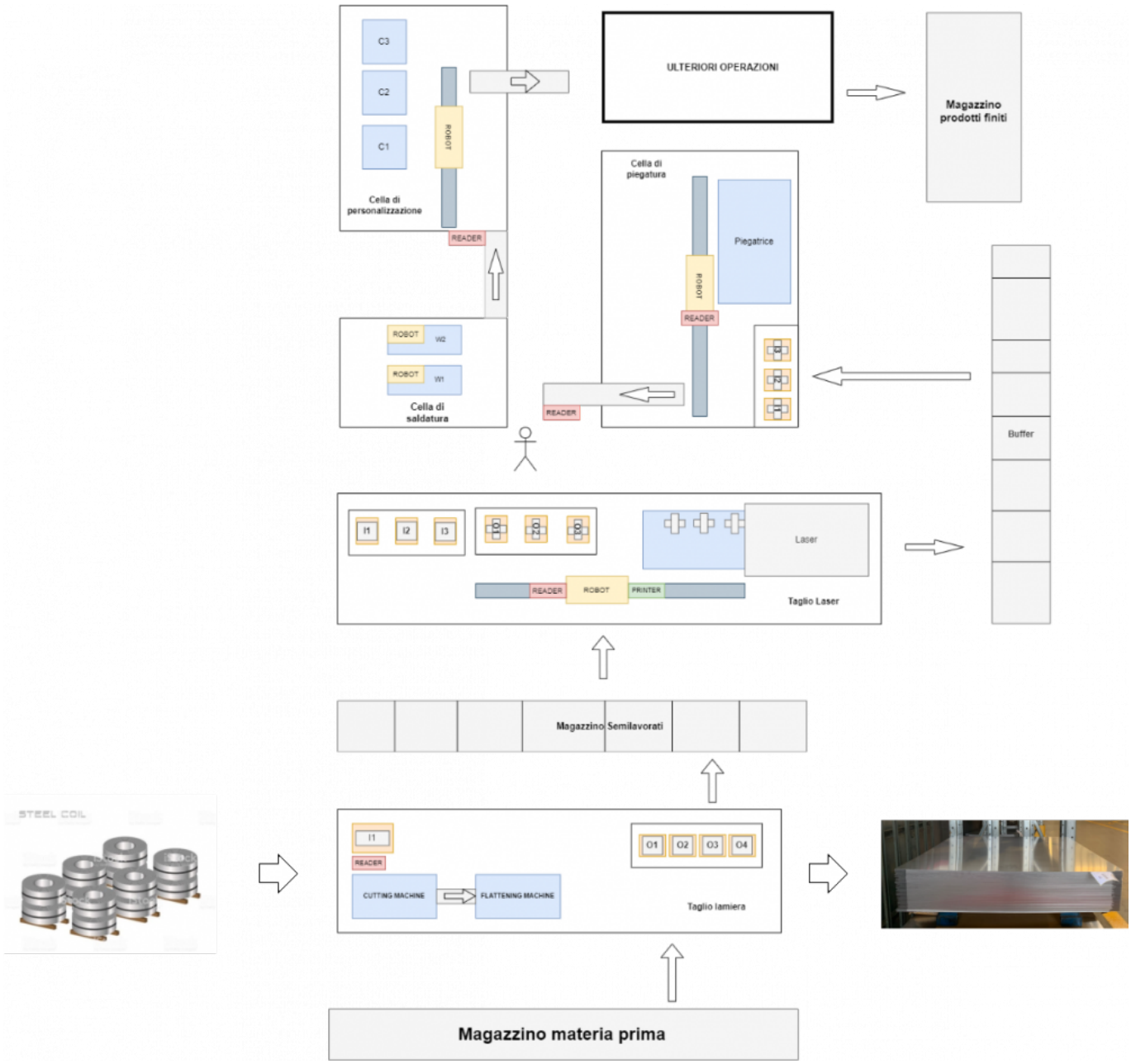
Di seguito la documentazione delle interfacce:

- Starmatik

Di seguito la documentazione delle interfacce:

- Documentazione

Processo Fac 40



App supporto magazzinieri

Aggiungere le seguenti tabelle

```
CREATE TABLE public.lotmap (
    codlot CHARACTER VARYING(51) NOT NULL,
    codlotp CHARACTER VARYING(51) NOT NULL,
    tm INTEGER,
    DATA CHARACTER VARYING(9),
    ora CHARACTER VARYING(9),
    codoper CHARACTER VARYING(51),
    codnote CHARACTER VARYING(51),
    qta DOUBLE PRECISION
);
COMMENT ON TABLE public.lotmap IS 'Mappa un lotto nuovo ad uno o piu'' lotti padri
da cui si e'' generato';
ALTER TABLE ONLY public.lotmap
```

```
    ADD CONSTRAINT codlot_codlotp_pkey PRIMARY KEY (codlot, codlotp);
ALTER TABLE ONLY public.lotmap
    ADD CONSTRAINT ix_lotmap_codlot FOREIGN KEY (codlot) REFERENCES
public.lottim(codice) ON UPDATE CASCADE ON DELETE CASCADE;
ALTER TABLE ONLY public.lotmap
    ADD CONSTRAINT ix_lotmap_codlotp FOREIGN KEY (codlotp) REFERENCES
public.lottim(codice) ON UPDATE CASCADE ON DELETE CASCADE;
CREATE TABLE public.lotprod (
    codlot CHARACTER VARYING(51) NOT NULL,
    codprod INTEGER NOT NULL
);
ALTER TABLE ONLY public.lotprod
    ADD CONSTRAINT ix_lotprod_codlot_codprod PRIMARY KEY (codlot, codprod);
ALTER TABLE ONLY public.lotprod
    ADD CONSTRAINT ix_lotprod_codlot FOREIGN KEY (codlot) REFERENCES
public.lottim(codice) ON UPDATE CASCADE ON DELETE CASCADE;
ALTER TABLE ONLY public.lotprod
    ADD CONSTRAINT lotprod_codprod FOREIGN KEY (codprod) REFERENCES
public.produbi(codice) ON UPDATE CASCADE ON DELETE CASCADE;
```

Viene costruito l'agente AgentLog40

operazione	request	parametro	ritorno
ask	baiestatus	rec(codubi) codice della macchina di cui si chiede lo stato	rec(stato,codop,codart,descrizio,npz,tnuovamis,npztot,call,alarmcall,codoper) ove stato = 0(mancanza materiale), 1(OK), 2(da caricare), 3(in caricamento) codoper = codice dell'operatore che eventualmente ha in carico la missione per la specifica macchina
ask	infopallet	rec(codlot,codoper) in input può arrivare il pallet o un pezzo del pallet per vedere le informazioni del pallet stesso, passa anche il codice dell'operatore connesso	rec(codlot,ruek,codart,descrizio,codpf,descpfnpz,coil,isempty,codubi,dubi,visevadi,abievadi) ruek = lista di ruek + codubi + dubi associati al pallet; gli attributi sono divisi da pipe, gli elementi da virgola codubi, dubi = location ove il pallet si trova, vuoto se non e' da nessuna parte visevadi = 1 se il codoper e' in qualche ubicaz.codoper con tiposet='P' e stato='B' altrimenti vale 0 abievadi = 1 se a) il pallet e' configurato con RUEK dello stesso codubib che ha la location assegnata all'operatore b) tutti i ruek richiesti siano inseriti c) se richiesto sia specificato il codpf altrimenti vale 0
ask	changepallet	rec(codlotold,codlotnew) palletold è il pallet da cui parte il trasferimento, palletnew è il pallet dove verrà spostato il materiale dell'old update giacese set codubi='codlotnew' where codubi='codlotold'; update lotprod set codlot='codlotnew' where codlot='codlotold'; update lottim set codart=OLDLOT.codart,codagg=OLDLOT.codagg where codice='newcodlot'; update lottim set codart=' ',codagg=' ' where codice='oldcodlot';	rec(OK) se tutto OK, rec(err) se ci sono errori
sapnis.ask	gettc (viene svolta ogni volta che si legge un ruek)	rec(rueck)	rec(dati_ruek,codubi) se trovato, rec(null) se non trovato codubi = codice del centro di lavoro associato

operazione	request	parametro	ritorno
ask	savepallet (crea la produbi e la associa al pallet; a produbi non viene avviata) 0) confronta il contenuto attuale del pallet con quello richiesto (codart + codpf + qta + ruek) 1) crei un lottim per ogni unità di qta mettendo anche il codpf (lottim.codagg) 2) depositi i lottim nel pallet	rec(codlot,ruek,codart,qta,codagg) ruek puo' avere piu' di un ruek divisi da virgola del tipo: ruek pipe codubi virgola ...	rec(OK) se tutto OK, rec(err) se ci sono errori
ask	initpallet	rec(codlot,codcoil) ruek puo' avere piu' di un ruek divisi da virgola	rec(OK) se tutto OK, rec(err) se ci sono errori
ask	confirmcall	rec(codubi,codoper) codice della macchina per la quale l'operatore prende la missione, operatore che prende la missione	rec(ok) se tutto OK, rec(err) se ci sono errori (oppure potrebbe essere richiamata la baiestatus
ask	evadimissione	rec(codubi,codlot) codice della macchina per cui si sta evadendo la missione, codice del pallet interessato	rec(ok) se tutto OK, rec(err) se ci sono errori; temporaneamente l'agente svolge i seguenti passi: 1) cerca una baia vuota della linea in ordine di pos 2) deposita nella prima baia vuota che trova 3) se non trova baie vuote da errore svolge lo stesso 4, 5 4) setta ubicaz.codoper='' e ubicaz.stato='A' per la macchina Nella versione definitiva saranno svolti solo i punti 4, 5

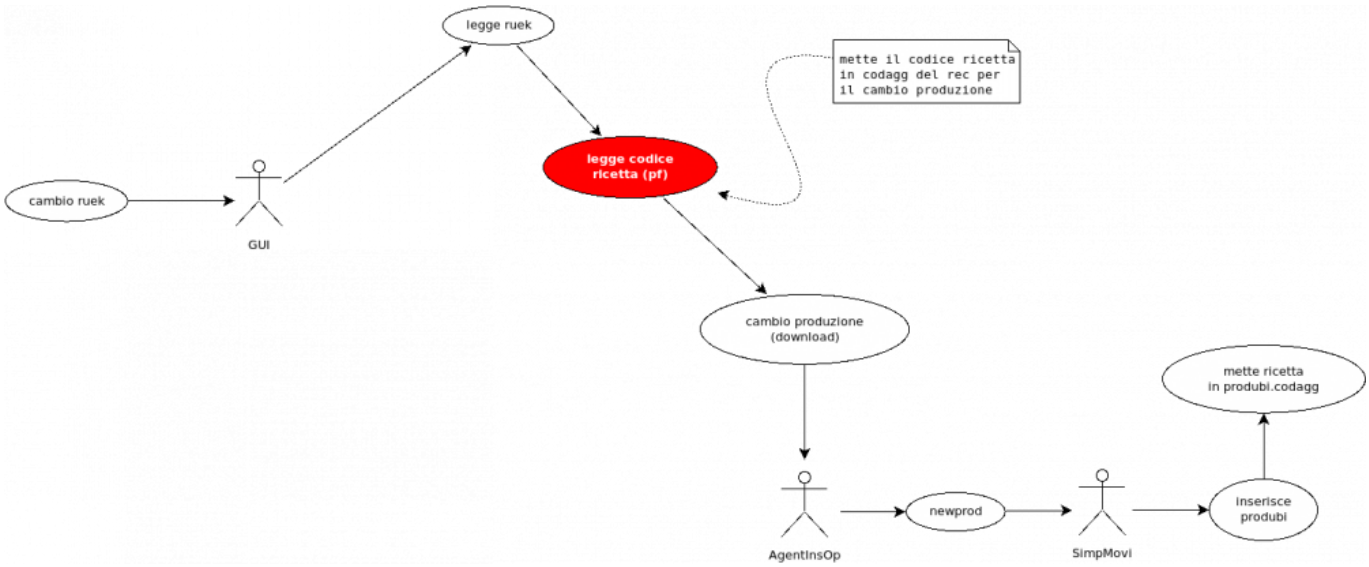
Al fine di identificare la baia in fase di lavorazione si utilizzerà il flag ubicaz.stato:

- V = vuota
- A = aperta (in lavorazione)
- C = chiusa (piena ma non in lavorazione)
- Nel caso di linea lo stato B ⇒ Chiamata presa in carico dal magazziniere, mentre ubicaz.codoper ⇒ operatore che ha preso la chiamata

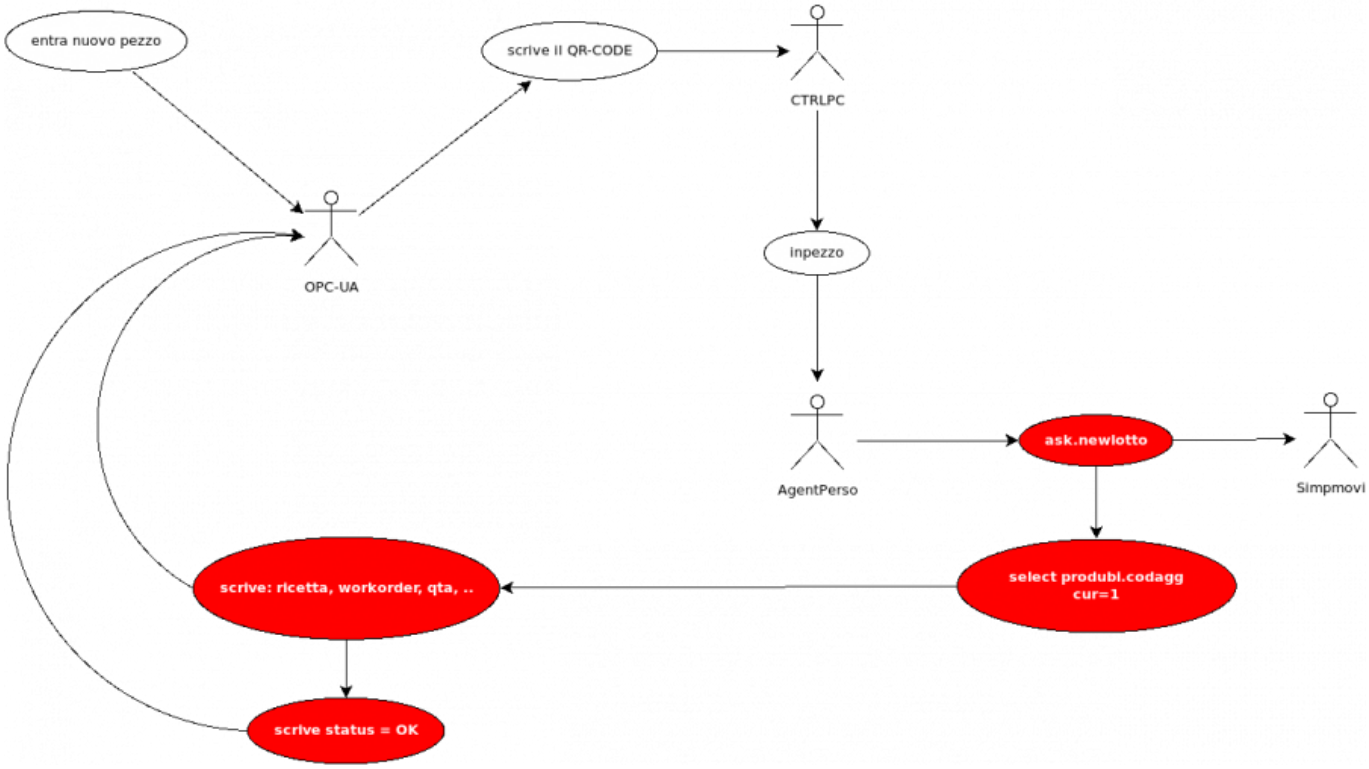
Personalizzazione

Flusso dati

Cambio produzione



Ingresso Pezzo



Dataset ingresso pezzo:

d1=codubi, d2="workorder"	
Nome campo	**Descrizio**
codlot	Codice della matricola in ingresso letta dal OPC e resettata da IMPROVE
codop	Workorder scritto da IMPROVE
codmod	Codice prodotto in lavorazione scritto da IMPROVE
qta	Quantità da produrre per il WO scritto da IMPROVE
codagg	Codice della ricetta scritto da IMPROVE

Dataset uscita pezzo:

d1=codubi, d2="exit"	
Nome campo	**Descrizio**
codlot	Codice della matricola in uscita letta dal OPC e resettata da IMPROVE

Dataset fermate stato macchine:

d1=codubi, d2="fermate"	
Nome campo	**Descrizio**
pres_status	Stato della macchina pressa valori 1=ready 2=work 3=not ready
bay_status	Stato della baia di carica e scarico valori 1=pronta 2=lavorazione 3=caricare scaricare
*_piece	Stato della presenza pezzo valori 0=non presente 1=presente
pres_oil	Stato olio del livello oil valori 0=KO 1=OK
robot_status	Stato del robot valori 1=Errore 2=stop 3=run

Piano attività

- Fine giugno impianto (Piegatura, Saldatura, Personalizzazione) installato in stabilimento:
  - **week 27** test con sistema installato in campo
- Primi giorni di maggio inizia installazione "Personalizzazione"
  - **week 19** test con sistema Personalizzazione
  - subito dopo inizierà a produrre con lotti di test
- Inizio giugno inizia installazione "Saldatura"
- **week 24** test con sistema di saldatura
- Interfaccia mulettista:
  - Verificare se era stato previsto uno smart device portatile oppure se si pensava di usare una postazione fissa
  - **week 20** app pronta per i test
- Touch next andon: piccola modifica sul hmi
  - studio interno
  - valutare possibilità di tracciare gli scarti associandoli ad un codice pallet
  - controllo qualità operatore: ***definire i dettagli***
- Dashboard:
  - ***definire i dettagli***
- Sinottico:
  - acquisire i layout dell'impianto (dxf)
  - ***definire i dettagli*** (es: scomporre l'impianto in piu' viste, visualizzare in linea su uno schermo il layout)
- Data acquisition:
  - variabili di processo
  - ***definire i dettagli***
- Perdite di disponibilità della macchina:
  - partire dai cluster di perdita e ***defini i dettagli***
- APES:
  - validare il dataset opc-ua (incontro)