



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA (DISI)

CORSO DI LAUREA IN INFORMATICA PER IL MANAGEMENT

BPMN nel settore manifatturiero: modellazione di processi produttivi tramite l'estensione ARMS

Tesi di laurea in Ingegneria del Software

Relatore:
Prof. Davide Rossi

Presentata da:
Riccardo Ventura

Sessione Dicembre 2025
Anno Accademico 2024/2025

INTRODUZIONE.....	2
1 INTRODUZIONE AL BPMN E CONTESTO MANIFATTURIERO.....	4
1.1 Lo standard BPMN 2.0: Storia, Scopi e Costrutti.....	4
1.1.1 Flow Objects.....	5
1.1.2 Connecting Objects.....	6
1.1.3 Swimlanes.....	6
1.1.4 Artifacts.....	6
1.2 BPMN nel Contesto Manifatturiero.....	7
1.3 Punti di forza di BPMN.....	8
2 LIMITI DEL BPMN STANDARD E NECESSITÀ DI ESTENSIONI.....	10
2.1 Criticità dei modelli BPMN nella pratica industriale.....	10
2.1.1 Problemi strutturali e logici.....	10
2.1.2 Problemi di layout e comprensibilità.....	10
2.1.3 Inconsistenze nell'Etichettatura.....	11
2.2 Il “Semantic Gap”: Dal modello grafico all'esecuzione della simulazione.....	11
2.3 L'invisibilità del mondo fisico: Flussi di materiali e Dispositivi IoT.....	12
2.3.1 Perdita del “Physical Object Flow”.....	12
2.3.2 Ambiguità nella modellazione IoT.....	12
2.4 Limiti nella simulazione quantitativa: Gestione a Lotti e Accuratezza.....	13
2.4.1 La sfida del Batch Processing Dinamico.....	13
2.4.2 Accuratezza dei risultati nel caso di studio Philips Healthcare.....	14
2.5 Verso una modellazione Resource-Centric: I requisiti mancanti.....	15
3 LO STATO DELL'ARTE E INTRODUZIONE AD ARMS.....	18
3.1 State of the Art.....	18
3.2 ARMS (Activity-Resource Modelling Simulator).....	18
3.3 Il Metamodello ARMS (Formalizzazione).....	19
3.4 Validazione Teorica: Come ARMS soddisfa i Requisiti.....	22
4 METODOLOGIA DI VALUTAZIONE.....	24
5 VALIDAZIONE SPERIMENTALE DEI REQUISITI.....	26
5.1 Flussi Materiali e Trasformazione Fisica (R1 + R6).....	26
5.2 Eterogeneità delle Risorse e Tempi Variabili (R8 + R9).....	28
5.3 Vincoli su Risorse Accessorie e Priorità (R5 + R7).....	30
5.4 Attività in Batch e Accumulo Dinamico (R2).....	32
5.5 Ottimizzazione Operativa (R3).....	35
5.6 Caso Studio Integrato.....	38
CONCLUSIONE.....	45
BIBLIOGRAFIA.....	46
RINGRAZIAMENTI.....	49

INTRODUZIONE

Il **Business Process Modeling and Notation** (BPMN 2.0) è lo standard globale per la modellazione dei processi aziendali e parte fondamentale del BPM (*Business Process Management*). Questo standard offre una notazione grafica comprensibile a tutti gli *stakeholders* coinvolti in un progetto o nelle attività aziendali, ottimizzando il *workflow* (sistema per la gestione di processi e attività ripetitive che si verificano in un particolare ordine), cosa che ha reso BPMN uno strumento fondamentale in svariati settori [\[1\]](#).

Tuttavia, l'applicazione dello standard ai processi puramente manifatturieri ha fatto emergere significative limitazioni. Infatti la notazione mostra evidenti carenze che generano un marcato *semantic gap* (divario semantico) tra il modello grafico e i requisiti operativi richiesti dai motori di simulazione, compromettendo l'accuratezza delle analisi quantitative.

Alla luce di tali criticità, la presente tesi si propone di superare i limiti dello standard attuale attraverso l'adozione dell'estensione **ARMS**, specificamente concepita per il settore manifatturiero. La prima fase dell'elaborato approfondirà le basi teoriche e le problematiche aperte, mentre la seconda si focalizzerà sulla modellazione dei processi in scenari produttivi concreti utilizzando l'estensione scelta.

Il lavoro è strutturato come segue:

- Il **Capitolo 1** fornirà una panoramica dello standard BPMN 2.0, contestualizzandolo nel settore manifatturiero e confrontandolo con altre notazioni esistenti per evidenziarne i punti di forza.
- Il **Capitolo 2** approfondirà le criticità del BPMN standard, analizzando nel dettaglio i limiti semantici che ne impediscono un uso efficace per la simulazione e la modellazione accurata dei processi produttivi e introdurrà i requisiti che un buon modello di simulazione che estende la notazione standard deve possedere.
- Il **Capitolo 3** presenterà lo stato dell'arte delle estensioni di BPMN e introdurrà formalmente ARMS come soluzione alle problematiche prima individuate.
- Il **Capitolo 4** definirà la metodologia di valutazione (approccio "**Atomic Testing**").

- Il **Capitolo 5** validerà l'estensione ARMS usando la metodologia definita nel capitolo precedente.

1 INTRODUZIONE AL BPMN E CONTESTO MANIFATTURIERO

1.1 Lo standard BPMN 2.0: Storia, Scopi e Costrutti

Come descritto in [2], la prima versione di BPMN nasce nel 2004 e già allora fu presentata come la notazione standard per la modellazione di processi produttivi. Seguirà poi la versione 1.1 (rilasciata nel 2008) ed infine, nel 2011, verrà pubblicata la versione attualmente in uso, la 2.0. L'obiettivo originale di BPMN era quello di offrire una notazione facilmente comprensibile a tutti gli attori aziendali coinvolti in un progetto [2]. Questo include gli analisti che elaborano la bozza iniziale del processo, gli sviluppatori incaricati dell'implementazione e i manager che lo dovranno poi monitorare e gestire.

Un altro fattore che ha contribuito allo sviluppo di BPMN è stata la necessità di eliminare la fase di traduzione tra il linguaggio di modellazione dei processi e il modello di esecuzione. Tali conversioni erano soggette a errori ma soprattutto rendevano difficile la comprensione dell'evoluzione e delle prestazioni dei processi da parte degli sviluppatori [2]. Per affrontare questo problema quindi gli sviluppatori di BPMN si sono imposti come obiettivo primario quello di creare un ponte tra la notazione visiva e i linguaggi di esecuzione.

Il focus per lo sviluppo della versione 2.0 [2] è stato quello di:

- Creare un metamodello, un formato dei diagrammi e una semantica di esecuzione standardizzati che consenta lo scambio di modelli di processi aziendali anche tra strumenti di diversi fornitori;
- Scrivere una notazione estesa per le interazioni inter-organizzative (anche note come coreografie di processo), che abilita nuovi casi d'uso per il supporto automatico degli strumenti per processi che coinvolgono diversi partner commerciali;
- Alcuni elementi di modellazione aggiuntivi per i processi, come eventi non-interrompibili e sotto-processi di evento, e per la modellazione di dati e archivi di dati;
- Una mappatura dettagliata da BPMN a BPEL (*Business Process Execution Language*), che dimostra l'allineamento di BPMN con gli strumenti e gli standard pre-esistenti per l'esecuzione di processi produttivi.

Queste aggiunte hanno portato al cambiamento del nome da “*Modeling Notation*” (BPMN 1.x) a “*Model and Notation*” (BPMN 2.0), rimarcando il fatto che BPMN nella nuova versione non sia più solo un insieme di simboli per disegnare un processo, ma uno standard completo per definire la logica, la struttura e l’esecuzione di un processo [2].

I costrutti della BPMN sono raggruppati in quattro categorie base:

- 1) *Flow Objects* (Oggetti di Flusso);
- 2) *Connecting Objects* (Oggetti di Connessione);
- 3) *Swimlanes* (Corsie);
- 4) *Artifacts* (Artefatti).

1.1.1 *Flow Objects*

I Flow Objects sono gli elementi principali che descrivono il comportamento di un processo. Si dividono in ulteriori tre tipi: *Activities*, *Events* e *Gateway* [2].

- **Activities (Attività)** [2]: L’attività è un’azione discreta, con un inizio e una fine ben definiti, eseguita come parte di un processo organizzato. Le attività possono essere ulteriormente dettagliate come **Task** (compiti) o **Subprocess** (sottoprocessi) che a loro volta contengono un flusso di attività. I **task** possono essere definiti come istanza singola, come *loop* (esecuzione sequenziale) o come istanze multiple (esecuzione parallela) e si classificano in:
 - **User task**: Compito svolto da un umano;
 - **Send/Receive task**: Compiti che rispettivamente inviano o attendono la ricezione di un messaggio;
 - **Script task**: Un compito la cui logica è definita da uno script;
 - **Service task**: Compito eseguito automaticamente da un servizio web.
- **Events (Eventi)** [2]: Gli eventi rappresentano qualcosa che accade durante un processo. Possono essere di tipo “*catching*” (cattura) se il processo risponde ad un innesco o di tipo “*throwing*” (lancio) se è il processo a generare un segnale. Si classificano in base alla loro posizione nel processo:
 - **Start**: Indicano dove inizia un processo;
 - **Intermediate**: Si verificano durante il flusso del processo, come eventuali errori o per inviare segnali;
 - **End**: Indicano dove finisce il processo.

- **Gateway (Porte Logiche)** [2]: Utilizzati per controllare la diramazione e l'unione dei percorsi all'interno del processo. Marcatori specifici indicano il loro comportamento (*or*, *exclusive-or*, *and* e *complex*).

1.1.2 *Connecting Objects*

Questi oggetti collegano i *Flow Objects* tra loro, definendo logica e sequenza del processo, si dividono in [2]:

- **Sequence Flow (Flusso di sequenza)**: È l'elemento principale di connessione. Definisce l'ordine di esecuzione delle attività all'interno di un singolo processo;
- **Message Flow (Flusso di Messaggio)**: Indica il flusso di messaggi e comunicazioni tra due entità di business separate;
- **Association (Associazione)**: Utilizzata per collegare informazioni o Artefatti ai *Flow Objects*.

1.1.3 *Swimlanes*

Le Swimlane sono costrutti grafici utilizzati per organizzare e categorizzare le attività in base ai partecipanti al processo, sono [2]:

- **Pool (Piscina)**: Rappresenta un partecipante principale nel processo e agisce da contenitore per le attività di quel dato partecipante (es. un cliente o un sistema);
- **Lane (Corsia)**: È una suddivisione interna di una Pool. Sono usate per organizzare le attività in base ai ruoli o funzioni specifiche all'interno dell'organizzazione (es. magazzino).

1.1.4 *Artifacts*

Gli Artefatti forniscono informazioni aggiuntive al processo senza influenzarne direttamente il flusso di esecuzione e sono [2]:

- **Data Object (Oggetto Dati)**: Mostra quali dati sono richiesti o prodotti da un'attività (es. un ordine o una fattura);
- **Text Annotation (Annotazione Testuale)**: Aggiunta di note o commenti esplicativi;

- **Group (Gruppo):** Raggruppamento visivo di elementi utilizzato per scopi di documentazione o analisi.

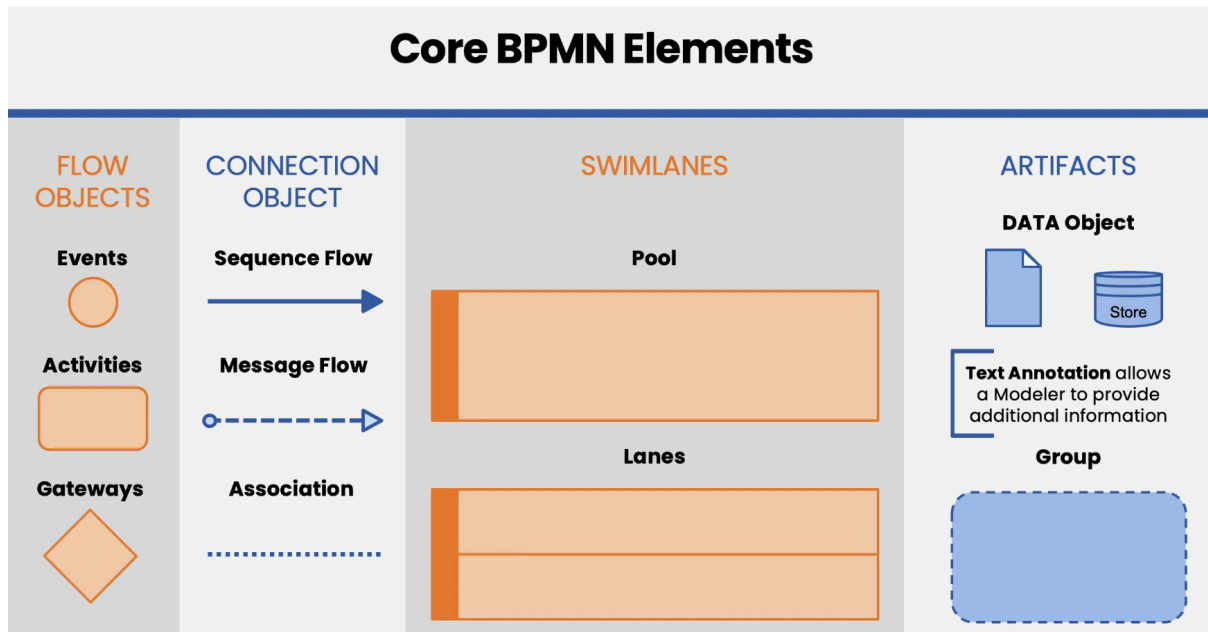


Fig.1: Costrutti primari di BPMN [3]

Infine, dato il notevole incremento dei concetti esprimibili tramite BPMN, tre livelli di utilizzo del linguaggio sono stati definiti [2]:

- **Livello 1: Modellazione Descrittiva** - orientata semplicemente alla documentazione del flusso di processo, principalmente per la comprensione e la comunicazione relative ai modelli “as-is” (stato attuale) e “to-be” (stato futuro);
- **Livello 2: Modellazione Analitica** - supporta l’analisi quantitativa e qualitativa rispetto agli indicatori chiave di performance;
- **Livello 3: Modellazione Eseguitibile** - modelli grafici che possono essere trasformati in specifiche basate su XML che guidano i motori di processo, rende possibile l’attivazione automatica dei modelli.

1.2 BPMN nel Contesto Manifatturiero

Le aziende manifatturiere, per studiare e migliorare le loro pratiche, hanno la necessità di visualizzare i loro processi da diversi punti di vista e a vari livelli di astrazione. Storicamente per questo scopo sono state utilizzate diverse notazioni, tra le quali **IDEF3** o **Value Stream Mappings** [4], a cui nel 2011 si è aggiunto lo standard **BPMN 2.0**.

Fin dalla sua introduzione, come discusso nel capitolo 1.1, BPMN 2.0 mostrava un ottimo potenziale per la standardizzazione delle interazioni di business grazie al suo scopo dichiarato di agire come “ponte” tra la progettazione e l’implementazione di processi produttivi, rendendolo un candidato ideale per l’integrazione dei sistemi informativi richiesta dall’Industria 4.0.

Tuttavia, prima di analizzare l’adeguatezza, i punti di forza e i limiti, è opportuno confrontarlo con le notazioni precedentemente menzionate (**IDEF3** e **VSM**), create per catturare aspetti che BPMN non considera nativamente:

- Rispetto a **IDEF3**, BPMN può essere visto come un “*superset*” degli schemi di processo IDEF3, aggiungendo un supporto esplicito per modellare i partecipanti al processo, i gestori di eventi e gli scambi di messaggi, andando a migliorare la prospettiva organizzativa di alto livello che BPMN offre. Tuttavia BPMN non è in grado di modellare gli oggetti esistenti e le loro transizioni, come possono fare gli schemi oggetto IDEF3 [\[4\]](#).
- Rispetto a **VSM** invece, che si configura come strumento ad alto livello per un’analisi rapida dei processi, BPMN si concentra sulla progettazione del processo con un livello di dettaglio superiore. In particolare, BPMN eccelle nella definizione della sequenza delle attività, permettendo così una migliore e più dettagliata implementazione dei processi stessi [\[4\]](#).

1.3 Punti di forza di BPMN

Si riassumono di seguito i principali punti di forza di BPMN emersi nei paragrafi precedenti:

- 1) **Gestione dei ruoli e interazioni tra entità diverse:** A differenza di altre notazioni, BPMN utilizza costrutti come *Pool* e *Lane* per permettere di categorizzare le attività in base al partecipante, rendendo immediatamente chiare le responsabilità organizzative all’interno del processo. Inoltre, grazie al flusso di messaggi, è possibile definire in dettaglio all’interno del modello le interazioni tra entità di business separate;
- 2) **Precisione esecutiva:** BPMN permette di definire la logica esatta del processo (tramite *gateway*) e la gestione puntuale degli eventi (come timer e errori), fondamentali per i sistemi informativi che supportano la produzione;

- 3) **Interoperabilità**: Obiettivo cardine della versione 2.0, che con il formato di serializzazione standardizzato (XML) ha facilitato lo scambio di modelli tra strumenti software diversi.

Nonostante questi pregi, BPMN presenta alcuni limiti, specialmente nell'applicazione ai processi puramente fisici del manifatturiero. Limiti che creano un **divario semantico** quando si tenta di utilizzare BPMN per simulare e ottimizzare le dinamiche complesse che presenta una linea di produzione. Argomento che sarà approfondito nel prossimo capitolo.

2 LIMITI DEL BPMN STANDARD E NECESSITÀ DI ESTENSIONI

2.1 Criticità dei modelli BPMN nella pratica industriale

Studi empirici su come gli utenti effettivamente utilizzano BPMN in contesti industriali reali hanno evidenziato lacune significative nella qualità dei modelli prodotti.

Un'analisi condotta su 585 modelli di processo BPMN 2.0 provenienti da sei diverse aziende [5] ha permesso di categorizzare le problematiche più frequenti in tre aree principali: **struttura**, **layout** ed **etichettatura**. Tali problematiche, all'apparenza solo di natura estetica, possono costituire reali impedimenti all'*automated processing of models*, compromettendo il corretto funzionamento dei motori di simulazione [5].

2.1.1 Problemi strutturali e logici

La correttezza strutturale è il prerequisito per l'interpretabilità logica di un processo. L'analisi sul campione di dati [5] evidenzia una presenza di *deadlocks* (situazioni di stallo che bloccano il processo) nel 22% dei processi e di *multimerges* (fusioni multiple non sincronizzate che portano a esecuzioni duplicate indesiderate nel flusso successivo) nel 42% di essi. Questi errori derivano spesso dal mancato utilizzo di *gateway* espliciti, una pratica che crea ambiguità tra logiche *AND* e *XOR* [5].

Evidenti problemi (che affliggono il 48% del campione) sono stati riscontrati nell'ambito dei flussi di messaggi, dove spesso vengono collegati a elementi errati o utilizzati impropriamente su eventi di lancio [5]. Ancora più problematica è la gestione dei *subprocesses*, dove nell'86% dei modelli i ruoli dei sottoprocessi non corrispondevano a quelli del processo principale, creando un'incosistenza che mina l'integrità dell'architettura di processo complessivo [5].

2.1.2 Problemi di layout e comprensibilità

L'analisi [5] ha mostrato che il 47% dei diagrammi eccedeva le dimensioni massime raccomandate (pagina A3), indicando un mancato uso di sottoprocessi o della scomposizione del modello in più parti per ridurre la complessità. Inoltre, il 27% dei modelli [5] presentava sovrapposizioni di archi o nodi e direzioni di flusso inconsistenti.

Un fenomeno tipicamente riscontrato è quello dei "modelli a banana", il cui flusso serpeggia da sinistra a destra per poi tornare indietro, violando la convenzione

temporale e spaziale della lettura, compromettendo la comprensibilità del diagramma [5].

2.1.3 Inconsistenze nell'Etichettatura

Circa il 42% delle etichette analizzate seguiva pattern sintattici ambigui o inconsistenti. È comune infatti trovare attività o eventi che non seguono le raccomandazioni delle linee guida (strutture verbo-oggetto per le attività e oggetto-participio per gli eventi) [5]. Infine, l'uso di glossari centralizzati per standardizzare i dati e i ruoli è quasi assente: circa il 72% dei ruoli e il 77% degli *data objects* non sono collegati ad una definizione formale, aumentando il rischio di interpretazioni errate dei dati scambiati durante il processo [5].

In conclusione, la pratica industriale dimostra che, senza linee guida rigorose e strumenti di validazione, i modelli BPMN tendono a degradare in qualità, aumentando i rischi in fase di modellazione di linee di produzione manifatturiere.

2.2 Il “*Semantic Gap*”: Dal modello grafico all'esecuzione della simulazione

Evidenziate le criticità riguardanti la correttezza sintattica e la leggibilità di un modello, un problema più profondo emerge quando si tenta di utilizzare un diagramma BPMN, anche formalmente corretto, come base per una simulazione quantitativa. In letteratura [6] questo problema è noto come ***semantic gap*** (divario semantico) e rappresenta la distanza tra un modello di processo business e la semantica operativa richiesta dai motori di simulazione.

Una criticità specifica di BPMN standard risiede nell'analisi “*ex-ante*” (pre-implementazione) delle prestazioni del processo produttivo sviluppato, ad esempio nella valutazione dei tempi di ciclo, nell'identificazione dei colli di bottiglia o nell'analisi dei costi di processo [6].

Per ovviare a questo problema, due tipi di approccio sono stati identificati [6]:

- 1) Approcci che utilizzano modelli di processo grafici estesi per la simulazione;
- 2) Approcci che si basano sull'idea di trasformare un modello di processo grafico in un modello di simulazione eseguibile.

2.3 L'invisibilità del mondo fisico: Flussi di materiali e Dispositivi IoT

Nel settore manifatturiero non ci si limita alla sequenza astratta delle attività, focus primario di BPMN, ma si lavora in particolare alla trasformazione e al movimento di entità fisiche (materie prime, semilavorati e prodotti finiti) e all'interazione con dispositivi hardware (sensori e attuatori). BPMN infatti mostra la sua natura orientata al processo, trascurando il flusso degli oggetti fisici.

2.3.1 Perdita del “*Physical Object Flow*”

Come discusso in letteratura [7], BPMN presenta una carenza assoluta nel trattare la semantica del flusso di oggetti (*object flow semantic*), e le sue intuizioni che dovrebbe fornire agli analisti e sviluppatori di sistema. Problema che si fa ancor più evidente quando si modellano i processi che trattano l'istanziamento del flusso di materiali, le cui informazioni non sono solo talvolta allegate agli oggetti, ma vengono anche modificate con ritardi, posizioni e trasformazioni che non possono essere completamente affrontate dalla notazione standard [7].

Ad esempio, un'attività denominata “Assemblaggio Componente” in BPMN standard descrive solamente l'azione logica dell'assemblaggio, omettendo tutto il processo di entrata, trasformazione e uscita del prodotto fisico che svanisce all'interno di costrutti non concepiti per tale scopo (come *task* o *data objects*)

Un veloce confronto con altre notazioni rende questa carenza ancor più manifesta. Gli *Activity Diagram* UML (*Unified Modeling Language*) ad esempio integrano nativamente i concetti di “*Object Flow*” e “*Object Semantics*” [7]. Tra i requisiti che UML ha soddisfatto nello studio [7] vi sono: le **informazioni sulla tracciabilità dinamica di articoli/lotti**, la **posizione effettiva**, il **campionamento per il controllo qualità** e l'**istanziamento delle operazioni di qualità**, la **quantità effettiva di scorte disponibili**, le **materie prime** e il **WIP** (*Work In Progress* - Lavoro in corso).

2.3.2 Ambiguità nella modellazione IoT

Con l'avvento dell'Industria 4.0, l'*Internet of Things* (IoT) ha reso i processi produttivi degli ibridi tra sistemi informativi e componenti fisici. In tale contesto emerge un ulteriore problema, ossia la scarsa percezione delle differenze tra *task* logici e *task* fisici [8].

Studi empirici [8] condotti sulla percezione dei modellatori (utenti) hanno dimostrato che l'utilizzo dei costrutti standard (in particolare *Service Task*, *Script Task* e *Business Rule Task*) per rappresentare azioni IoT genera confusione e ambiguità. Un utente che deve rappresentare l'azione di un sensore o un attuatore è costretto ad usare gli stessi simboli grafici impiegati per operazioni puramente software.

La ricerca [8] evidenzia due tipi di percezione errata:

- **Visiva:** le icone standard non sono sufficienti per comunicare la natura fisica dell'attività (i partecipanti non sono stati in grado di individuare la presenza di sensori o attuatori basandosi solo sulla notazione grafica);
- **Etichettatura:** anche l'uso di etichette testuali esplicite non garantisce una corretta interpretazione.

Tale mancanza di chiarezza comporta un aumento del carico cognitivo e il rischio di compromettere la qualità di implementazione dei processi.

2.4 Limiti nella simulazione quantitativa: Gestione a Lotti e Accuratezza

Dopo aver analizzato le incongruenze strutturali e l'invisibilità del flusso fisico degli oggetti, è necessario esaminare come queste carenze semantiche si traducano in errori quantitativi durante l'esecuzione di una simulazione. Nel contesto manifatturiero, la validità di un modello non si misura solo sulla sua correttezza semantica, ma sulla capacità di prevedere fedelmente i *Key Performance Indicators* (KPI), quali *throughput time* (tempo necessario perché un'unità di lavoro attraversi l'intero processo) e tassi di utilizzo delle risorse. Tuttavia, la rigidità dei costrutti BPMN standard di fronte a logiche produttive dinamiche, come il *batch processing* (gestione a lotti) [9], porta spesso a risultati inattendibili.

2.4.1 La sfida del *Batch Processing* Dinamico

Una *batch processing activity* [9] è considerata un'attività nella quale la risorsa elabora più istanze di un oggetto contemporaneamente in un'unica esecuzione dell'attività (ad esempio, un forno industriale che cuoce più componenti simultaneamente). La gestione a lotti e gli aspetti dinamici del cambiamento delle capacità di un'attività nel tempo diventano un problema serio quando si eseguono simulazioni di modelli di processo complessi, specialmente in quelli che presentano diversi livelli gerarchici (sottoprocessi) [9].

Sebbene BPMN permetta di modellare istanze multiple, fatica a replicare scenari dove gli oggetti sono processati dinamicamente e il loro numero cambia nel tempo. Infatti emergono degli errori di simulazione quando la capacità di un'attività non è fissa ma stocastica o dipendente dal flusso precedente e il processo presenta *gateways* che dividono il flusso [9].

Non viene inoltre considerata la variabilità degli scenari operativi adottati dalle aziende per l'esecuzione delle attività: alcune attività vengono eseguite più volte per input multipli, alcune vengono eseguite una sola volta per istanze multiple e alcune attività vengono eseguite più volte per una singola istanza di input. Questi scenari influiscono anche sulla durata della *batch processing activity* nella simulazione del processo e devono essere gestiti [9].

2.4.2 Accuratezza dei risultati nel caso di studio *Philips Healthcare*

L'incapacità di modellare accuratamente vincoli complessi e comportamenti di gruppo ha conseguenze dirette sull'accuratezza dei dati in uscita. Uno studio comparativo condotto sul processo di ricondizionamento presso Philips Healthcare [10] ha messo a confronto i risultati di una simulazione eseguita con un tool BPMN standard (Signavio) e un software specializzato (Arena), evidenziando come il *semantic gap* del BPMN porti ad una rappresentazione eccessivamente ottimistica della realtà produttiva. Viene infatti segnalata una sottostima del 33% del *throughput time* e del 6% del tasso di utilizzo reale da parte della simulazione standard.

Ciò viene attribuito specialmente a 3 aspetti individuati in [10]:

- **”Interleaved Parallel Routing”**: situazione che richiederebbe il vincolo per cui una sola *branch* (ramo) del processo possa lavorare mentre le altre restano in attesa (interpretata da Signavio come semplice *gateway* parallela, dove tutti i *task* potevano essere eseguiti simultaneamente);
- **Aggiunta di risorse aggiuntive** per un *task*, cosa che in Signavio non è possibile fare (quindi le altre risorse sono libere di svolgere gli altri compiti);
- **Task di approvazione**, dove l'avanzamento dell'attività si basa sulla decisione presa dalla risorsa (persona) che approva il compito. Tutti questi *pattern* comportano un aumento del *throughput time* e una diminuzione dello *usage rate*, giustificando così le discrepanze riscontrate nei risultati delle simulazioni effettuate con Signavio.

2.5 Verso una modellazione *Resource-Centric*: I requisiti mancanti

Per essere veramente utile in ambito manifatturiero, e quindi permettere di catturare adeguatamente le dinamiche legate alle risorse e ai materiali, BPMN, attraverso un'estensione, deve cambiare la sua struttura da **Activity-centric** (focus sulle attività) a **Resource-centric** (focus sulle risorse) [11], ponendo al centro della modellazione non solo cosa deve essere fatto, ma chi (o cosa) lo esegue e con quali vincoli.

Basandosi su un'analisi della letteratura scientifica è stato possibile formalizzare una griglia di **nove requisiti fondamentali (R1-R9)** che un metamodello destinato alla simulazione manifatturiera deve soddisfare [11].

R1: Combinazione e Separazione dei Prodotti

Sebbene il BPMN gestisca la convergenza dei flussi tramite *gateway*, manca di costrutti specifici per descrivere la logica fisica di assemblaggio (combinazione) e disassemblaggio (separazione). Nella manifattura è essenziale modellare come diverse materie prime o semilavorati convergono per formare un nuovo prodotto unico o viceversa e le estensioni che usano *fork* e *join* non riescono a catturare i dettagli delle trasformazioni fisiche del prodotto.

R2: Attività in Batch

Come affrontato nel paragrafo 2.4.1, BPMN non possiede nativamente un meccanismo che trattiene gli elementi (*item*) fino al soddisfacimento di specifiche condizioni (ad esempio, un accumulo di 15 pezzi) prima di avviare l'attività in modo dinamico.

R3: Affinità dell'Esecutore

In molti contesti industriali un singolo esecutore può eseguire diverse attività sulla stessa risorsa per aumentare l'efficienza (ad esempio, un operatore che esegue dei controlli di qualità subito dopo aver assemblato quel determinato pezzo). BPMN non consente di esprimere questo vincolo di "affinità", rischiando di assegnare il *task* successivo ad una risorsa libera qualsiasi, ignorando le reali logiche di ottimizzazione operativa.

R4: Assegnazione basata sui parametri dell'Esecutore

Spesso la selezione di una risorsa è guidata da parametri specifici come il costo di avviamento o il suo stato attuale. È necessario quindi un requisito che permetta di selezionare l'esecutore migliore basandosi sulle sue proprietà intrinseche, una logica che BPMN non prevede.

R5: Dipendenze Multi-Esecutore

È comune che una data attività venga eseguita da un esecutore utilizzando una o più risorse insieme (ad esempio, un operatore che usa una lavatrice e alcuni detersivi) o da diversi esecutori che lavorano insieme (ad esempio, una linea di assemblaggio con più operatori). È dunque necessario poter esprimere la cardinalità e la tipologia di tali dipendenze, bloccando l'attività se il set completo di risorse non è disponibile.

R6: Gestione dell'Inventario

Il concetto di *Inventory* (magazzino o stock) è assente in BPMN, che vede il processo come un flusso continuo. Tuttavia, la gestione delle scorte, dei semilavorati invenduti e delle materie prime disponibili è un prerequisito per l'avvio delle attività produttive.

R7: Priorità delle Attività

Quando risorse limitate sono contese da più processi o *task* simultanei, è necessario un sistema decisionale per risolvere il conflitto. Questo requisito impone la possibilità di definire dei livelli di priorità tra le attività, permettendo quindi al simulatore di prendere una scelta in base alle politiche aziendali.

Oltre ai requisiti provenienti dall'analisi della letteratura, l'interazione con gli esperti di dominio ha fatto emergere due ulteriori necessità critiche [\[11\]](#):

R8: Relazione tra Attività, Tipo di Prodotto ed Esecutore

Non tutti gli esecutori possono lavorare su tutti i prodotti. Esiste spesso una matrice di compatibilità rigida: un determinato prodotto può essere lavorato solo da una specifica macchina o da un operatore con una certificazione particolare. Questo vincolo ternario (Attività, Tipo di Prodotto, Esecutore) deve poter essere modellabile in modo esplicito.

R9: Durata variabile basata sull'Esecutore

La durata di un'attività è spesso basata su *chi* la esegue. Ad esempio, è verosimile che un macchinario di nuova generazione sia più efficiente rispetto ad uno più obsoleto, oppure la performance di un esecutore potrebbe essere influenzata da fattori esterni. Il modello quindi deve essere in grado di disaccoppiare il tempo di esecuzione dall'attività in sé, legandolo invece alla specifica coppia attività-esecutore.

La soddisfazione di questi nove requisiti definiti in [\[11\]](#) costituisce la base teorica su cui è stata progettata l'estensione **ARMS** [\[12\]](#), che verrà formalmente definita e analizzata nel capitolo successivo.

3 LO STATO DELL'ARTE E INTRODUZIONE AD ARMS

3.1 *State of the Art*

Definiti i requisiti che una buona estensione di BPMN per il manifatturiero deve possedere, si procede con una breve analisi di cinque estensioni (tra le numerose presenti in letteratura) che toccano ambiti diversi del settore manifatturiero, evidenziandone però l'insufficienza nel coprire i requisiti precedentemente descritti.

Le estensioni prese in esame (e il loro ambito) sono:

- 1) **Zor et al.** (Flusso Materiale e Logistica) [\[13\]](#);
- 2) **Manufacturing Process Modeling Notation** (Simulazione ed Esecuzione) [\[14\]](#);
- 3) **Inter-Organizational Business Processes for Industry 4.0** (Collaborazione Inter-Organizzativa) [\[15\]](#);
- 4) **Industry 4.0 Process Modeling Language** (Sistemi Ciber-Fisici e Architettura) [\[16\]](#);
- 5) **BPMNE4IoT** (IoT e Connettività) [\[17\]](#).

Analizzando le cinque estensioni, emerge un quadro dove il panorama di ricerca attuale è frammentato in programmi specializzati, ognuno dei quali risolve una parte del problema ma ne trascura altri cruciali per una simulazione manifatturiera completa.

Sebbene ciascuna estensione soddisfi in varia misura alcuni requisiti, la terna **R3-R8-R9** (Affinità Esecutore, Relazione “Attività-Tipo di Prodotto-Esecutore” e Durata Variabile) risulta costantemente trascurata, infatti in nessuna estensione è presente un modello di risorsa dinamico dove un esecutore ha uno stato, capacità specifiche e una matrice di compatibilità con specifici tipi di prodotto, impedendo una corretta simulazione che mimi le complesse decisioni di scheduling di una fabbrica reale.

3.2 **ARMS (Activity-Resource Modelling Simulator)**

Alla luce delle criticità emerse dall'analisi dello stato dell'arte in relazione ai nove requisiti definiti nel capitolo 2.5, la soluzione adottata in questo elaborato è **ARMS** (*Activity-Resource Modelling Simulator*) [\[12\]](#).

L'obiettivo primario di ARMS è quello di colmare il divario semantico tra la modellazione orientata alle attività (**activity-centric**), tipica di BPMN, e la simulazione orientata alle risorse (**resource-centric**), necessaria per un'analisi quantitativa accurata dei processi produttivi [12]. Tale approccio mira a rendere la simulazione dei processi significativamente più comprensibile e standardizzata rispetto all'uso di simulatori basati sulla *Discrete Event Simulation* (DES), i quali richiedono competenze specifiche per un utilizzo corretto [12].

La filosofia alla base di ARMS, quindi, è quella di non considerare questi due approcci come mutualmente esclusivi ma di integrarli in un unico metamodello, arricchendo l'approccio *activity-centric* con concetti derivanti dalla visione *resource-centric*, eliminando contestualmente la necessità di scrivere codice *ad-hoc* e riducendo la barriera di ingresso [12].

L'architettura di ARMS si compone di due elementi software complementari [12]:

- 1) **Modeler**: Un ambiente di modellazione basato su *React* e *BPMN-js* che consente agli utenti di definire il processo con nuove *feature* costruite sulla base dello standard BPMN;
- 2) **Simulation Engine**: Un motore di simulazione a eventi discreti basato sulla libreria *Kalasim* capace di interpretare il file generato dal modeler ed eseguire la simulazione, restituendo dati approfonditi sulle performance del processo.

Nei prossimi capitoli verrà formalizzato il metamodello su cui si basa ARMS, dettagliando le entità introdotte, e come queste soddisfano i requisiti di modellazione manifatturiera identificati nel capitolo 2.5.

3.3 Il Metamodello ARMS (Formalizzazione)

Per rendere operativa l'integrazione della prospettiva *resource-centric* all'interno di un flusso BPMN standard (*activity-centric*), è stato definito un metamodello formale di ARMS che estende le entità base del BPMN con nuovi concetti specifici per il dominio manifatturiero [12].

Il metamodello si compone di **elementi strutturali** (definiti in fase di modellazione) ed **elementi dinamici** (istanziati a *runtime* durante la simulazione). Di seguito vengono descritte le sei entità cardine che costituiscono la struttura dell'estensione ARMS [12]:

- 1) **Activity**: Sostanzialmente allineato all'entità "Attività" di BPMN standard con l'aggiunta dell'attributo *priority* che ne determina la priorità. Le *Activity* possono essere raggruppate ed eseguite insieme in determinate condizioni (ad esempio, una lavatrice industriale che è in grado di lavare 15 oggetti alla volta).
- 2) **Executor**: Rappresenta qualsiasi risorsa attiva (umana o robotica) impegnata nell'esecuzione di un'attività all'interno del processo produttivo, assimilabile all'entità *Performer* di BPMN standard. Ogni *Executor* è caratterizzato dall'attributo *quantity*, che specifica il numero di unità disponibili con le medesime caratteristiche.
- 3) **Product**: Rappresenta ogni singolo oggetto che subisce le trasformazioni durante il processo produttivo (il cui ciclo di vita è gestito dal simulatore). Inoltre, per permettere la modellazione dei diversi tipi di prodotti trattati nel processo (ad esempio, materie prime diverse o semilavorati), viene utilizzata l'entità statica *ProductFamily* che definisce la tipologia a cui i singoli prodotti appartengono.
- 4) **Accessory**: Modella le risorse passive o gli strumenti ausiliari necessari affinché un *Executor* possa completare il proprio *task*. Anche questa entità possiede l'attributo *quantity*, che specifica il numero di unità disponibili con le medesime caratteristiche.
- 5) **WorkItem**: Elemento dinamico generato all'attivazione di un *task* dove uno o più *Executors* sono stati identificati come potenziali esecutori tramite criteri di compatibilità. Quando specifici *Executors* vengono assegnati agli *WorkItem* un'istanza *Assignment* è creata.
- 6) **Inventory**: Rappresenta la quantità di prodotti disponibili dalle quali le attività possono attingere, come un deposito.

Descritte le entità fondamentali di ARMS [\[12\]](#), si illustrano di seguito le relative associazioni:

- **Compatibility** (*Activity-ProductFamily-Executor*): È frequente che alcuni *task* possano essere svolti esclusivamente da una specifica macchina o da operatori specializzati. *Compatibility* permette di descrivere quale *Executor* può lavorare su quella determinata *ProductFamily* in una certa *Activity*. Questa entità include inoltre informazioni sulla durata dell'operazione, in

quanto diversi *Executor* potrebbero non impiegare lo stesso ammontare di tempo a svolgerla.

- **Compatibility-Accessory:** Cattura la relazione e la cardinalità di un'attività svolta da un esecutore che utilizza una o più risorse insieme (ad esempio, quando usa una lavatrice industriale e qualche detergente) o da diversi esecutori che lavorano congiuntamente (ad esempio, una catena di montaggio con più operatori).
- **Assignment:** Rappresentazione della storia degli incarichi, descrivendo quindi chi (o cosa) ha svolto quella determinata attività. Particolarmente utile per aumentare l'efficienza complessiva del processo produttivo, in quanto è possibile determinare se un esecutore è in grado di svolgere ulteriori operazioni sul prodotto su cui ha appena lavorato (ad esempio, un controllo di qualità).
- Per spiegare come i prodotti vengono trasformati durante il processo, due associazioni vengono in aiuto: la "auto-associazione" in *ProductFamily* e **Transformation** (*Activity-ProductFamily*). La prima permette di descrivere da quali oggetti viene prodotto un oggetto e in cosa si può trasformare a sua volta. La seconda invece indica in quale attività un prodotto subisce una determinata trasformazione. Inoltre, poiché più trasformazioni possono avvenire nella stessa attività, in ARMS è stata adottata una cardinalità molti-a-molti in *Consume* e *Produce*; grazie a tali relazioni è possibile modellare scenari in cui è necessario attendere un determinato numero di oggetti prima che un'attività sia pronta ad essere eseguita.
- **Inventory-Product:** serve a designare in quale inventario un determinato prodotto può essere immagazzinato.

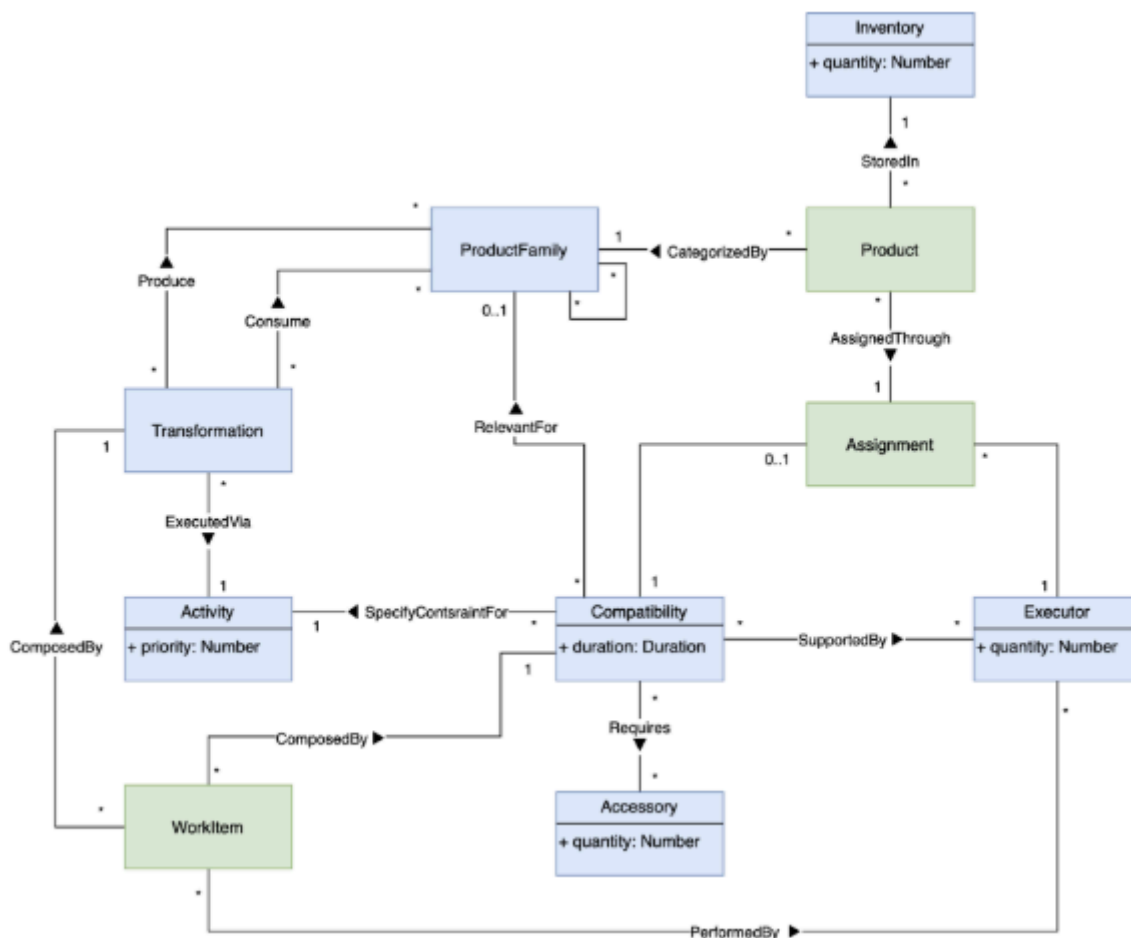


Fig.2: Diagramma del metamodello [12]

3.4 Validazione Teorica: Come ARMS soddisfa i Requisiti

Si dimostra la validità del metamodello ARMS [12] verificando come le entità sopra definite rispettano i nove requisiti (R1-R9) identificati nel Capitolo 2.5 come lacune di BPMN standard nell'ambito della modellazione di processi produttivi nel settore manifatturiero.

La seguente tabella riassume la mappatura tra i requisiti del dominio manifatturiero e le soluzioni implementate nel metamodello ARMS [12]:

Requisito	Soluzione in ARMS	Dettagli Implementativi
R1: Combinazione e Separazione dei Prodotti	Entità Transformation	Permette di definire input multipli (assemblaggio) e output multipli (disassemblaggio) per una singola attività, gestendo la logica di consumo e produzione.

R2: Attività in Batch	Entità <i>Transformation</i>	L'attributo <i>quantity</i> su <i>consume</i> e <i>produce</i> nella trasformazione permette di bloccare l'attività finché non è disponibile il numero di pezzi necessario.
R3: Affinità dell'Esecutore	Entità <i>Assignment</i>	Traccia la storia delle esecuzioni permettendo al simulatore di assegnare lo stesso esecutore a prodotti che ha già lavorato in precedenza.
R4: Assegnazione basata sui parametri dell'Esecutore	Entità <i>Executor</i>	Gli <i>Executor</i> sono istanze con parametri propri; il simulatore può interrogarli per selezionare la risorsa migliore.
R5: Dipendenze Multi-Esecutore	Associazione <i>Compatibiliy-Accessory</i>	È possibile modellare che un <i>Executor</i> , per essere compatibile con un task, necessiti obbligatoriamente di una o più risorse accessorie
R6: Gestione dell'Inventario	Entità <i>Inventory</i>	Traccia la quantità di prodotti disponibili da cui le attività possono attingere
R7: Priorità delle attività	Attributo <i>priority</i> in <i>Activity</i>	Gestisce i conflitti quando più processo competono per le stesse risorse limitate
R8: Relazione tra Attività, Tipo di Prodotto ed Esecutore	Entità <i>Compatibility</i>	La matrice di compatibilità esplicita formalmente quali combinazioni di <i>Executor</i> e <i>ProductFamily</i> sono valide per una data attività
R9: Durata variabile basata sull'Esecutore	Attributo in <i>Compatibility</i>	Il tempo di esecuzione non è più una proprietà fissa del <i>task</i> (come in BPMN standard), ma è definito nell'entità <i>Compatibility</i> , variando quindi in base all'esecutore scelto.

4 METODOLOGIA DI VALUTAZIONE

Definito formalmente il metamodello ARMS e averne dimostrato la validità teorica rispetto ai requisiti **R1-R9**, questo capitolo funge da ponte verso la fase sperimentale della tesi (dettagliata nel Capitolo 5). L'obiettivo è delineare la metodologia utilizzata per verificare sul campo l'efficacia dell'estensione utilizzata.

Sebbene una validazione basata esclusivamente sulla replica di casi di studio complessi ("*As-Is*" e "*To-Be*" su interi processi industriali) rappresenti solitamente lo standard per dimostrare l'applicabilità nel mondo reale, in questo lavoro si è optato per una strategia di "**Atomic Testing**". Questa metodologia prevede la creazione di micro-modelli *ad-hoc*, sviluppati *ex-novo*, ciascuno focalizzato sull'isolamento e la validazione di uno o due specifici requisiti tra i nove identificati. Invece di tentare di forzare la simulazione di processi complessi all'interno di Signavio — operazione che spesso richiede pesanti compromessi semantici — si procederà confrontando come ogni singolo costrutto viene gestito nello standard BPMN (evidenziandone i limiti) rispetto alla soluzione offerta da ARMS; ciò non verrà applicato per **R4**, visto che il concetto di *Executor* è intrinseco e obbligatoriamente presente in tutti i modelli, non necessitando quindi di un modello specifico.

Tale metodologia è motivata da tre fattori critici:

- 1) **Isolamento delle Variabili e Chiarezza Teorica:** Dove l'utilizzo di scenari complessi rischierebbe di distogliere l'attenzione sulla specifica carenza semantica, l'uso di micro-modelli specifici permette di isolare la singola o la coppia di variabili, rendendo la dimostrazione del "divario semantico" più immediata.
- 2) **Il problema di Signavio e dei modelli presentati negli studi:** Spesso i modelli raffigurati nella letteratura non sono simulabili in quanto BPMN standard (e quindi Signavio) ha una rigorosità particolarmente elevata che non permette la corretta rappresentazione delle logiche produttive dinamiche, dovendo quindi andare a modificare il processo con pesanti *workaround* grafici oppure lasciarlo incompleto, compromettendone la simulazione.
- 3) **Stato dello sviluppo di ARMS:** come descritto nel capitolo "*Conclusions and future works*" di [\[12\]](#) lo sviluppo di ARMS deve ancora essere completato. Elementi come *sub-processes*, *groups* e *pools* di BPMN standard non sono

ancora stati implementati, rendendo la modellazione di processi con gerarchia complessa difficoltosa da attuare.

A seguito della validazione dei requisiti implementati, verrà replicato un ipotetico processo produttivo più articolato al fine di dimostrare il potenziale di ARMS nella modellazione di scenari produttivi reali.

5 VALIDAZIONE SPERIMENTALE DEI REQUISITI

5.1 Flussi Materiali e Trasformazione Fisica (R1 + R6)

Per validare la capacità del sistema di gestire la convergenza di materiali fisici e il loro stoccaggio, è stato ideato un processo elementare dove un operatore assembla un tavolo, per creare questo tavolo sono necessarie **quattro gambe** e **un piano** (vincolo di Trasformazione: **R1**), ogni componente avrà un inventario specifico e l'attività non potrà iniziare se non sono fisicamente presenti in *stock* tutti i componenti necessari alla costruzione, inoltre dopo il compimento del *task* le scorte di componenti devono decrescere mentre quelle dei prodotti finiti devono crescere (consumo e produzione; vincolo di Inventario: **R6**).

Modellazione As-Is

Questo è un modello che potrebbe replicare lo scenario:

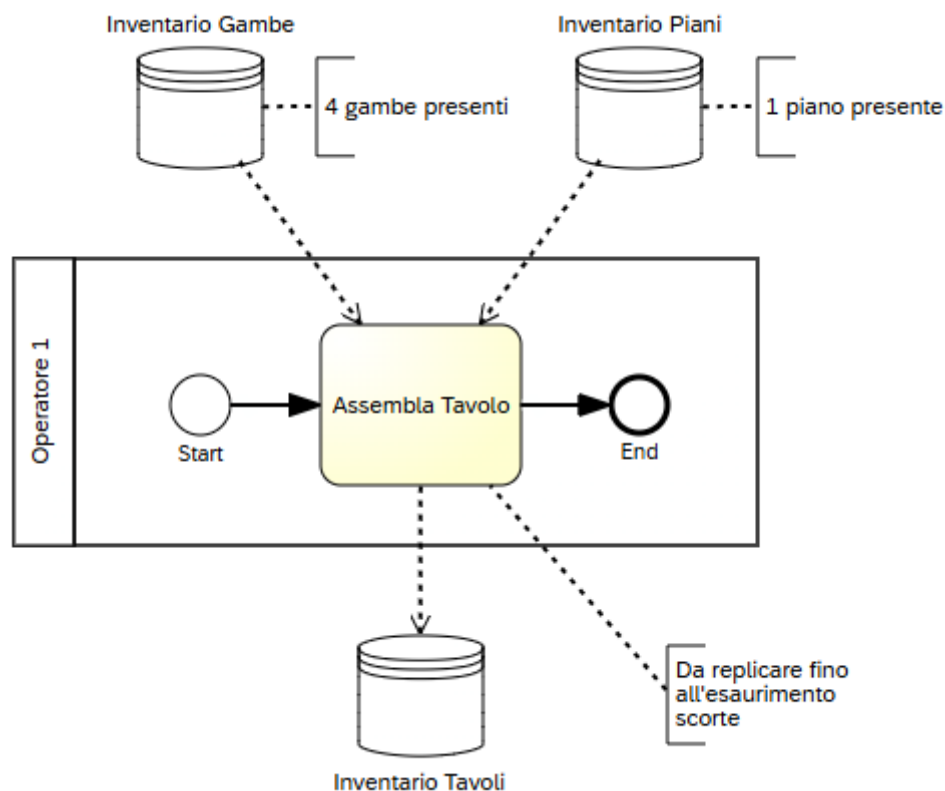


Fig.3: Scenario "Assemblamento Tavolo" in Signavio

Criticità emerse:

- 1) **Assenza di Logica Combinatoria (Fallimento R1):** Il BPMN standard non tratta i *Data Store* come magazzini fisici e non esiste un costrutto nativo per istruire il motore di simulazione a prelevare una quantità specifica di *token* da input multipli per generare un unico output;
- 2) **Simulazione Ottimistica (Fallimento R6):** In Signavio, la simulazione verifica solo se il flusso raggiunge l'attività, non c'è un reale controllo sulla disponibilità dello stock. Se nel magazzino virtuale ci fossero 3 gambe, la simulazione standard potrebbe comunque procedere ignorando il vincolo fisico e falsando i dati di *throughput*.

Modellazione To-Be

Replichiamo lo stesso scenario in ARMS:

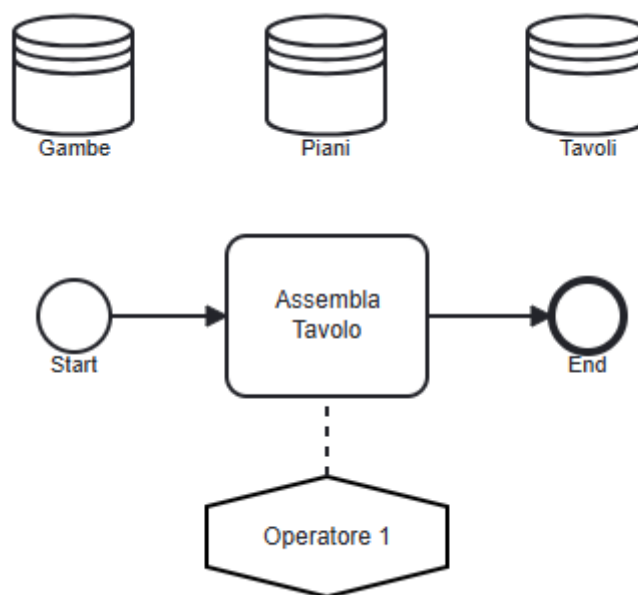


Fig.4: Scenario "Assemblamento Tavolo" in ARMS

Risultato della Validazione

Il simulatore monitora lo stato degli inventari, se ha a disposizione i materiali stabiliti nella *Transformation* all'interno dell'attività "Assembla Tavolo" (in questo caso quattro gambe e un piano per costruire un tavolo), procede con l'esecuzione. In caso contrario, qualora i materiali non siano sufficienti o la domanda di tavoli assemblati superasse le scorte a disposizione, il processo si interrompe, impossibile da replicare nativamente nel modello *As-Is* (rispetto di **R1** e **R6**).

5.2 Eterogeneità delle Risorse e Tempi Variabili (R8 + R9)

In questo scenario si vuole dimostrare come il sistema gestisce una situazione comune in fabbrica: lo stesso compito può essere svolto da macchine diverse che hanno prestazioni e capacità differenti. Si simula quindi un processo di stampa 3D di un componente in cui due stampanti diverse, una stampante A veloce ed efficiente (con durata dell'attività di 10 minuti) e una stampante B lenta e limitata (20 minuti), sono in grado di fare lo stesso tipo di lavoro (vincolo di Durata: **R9**). A ciò aggiungiamo la possibilità di stampare i componenti in due diversi tipi di materiale, come la plastica standard e la plastica rinforzata, e che quest'ultima è in grado di essere processata solamente dal primo tipo di stampante ma che invece dei soliti 10 minuti ne impiega il doppio (vincolo di Compatibilità: **R8**).

Modellazione As-Is

Possibile modello che replica lo scenario appena descritto:

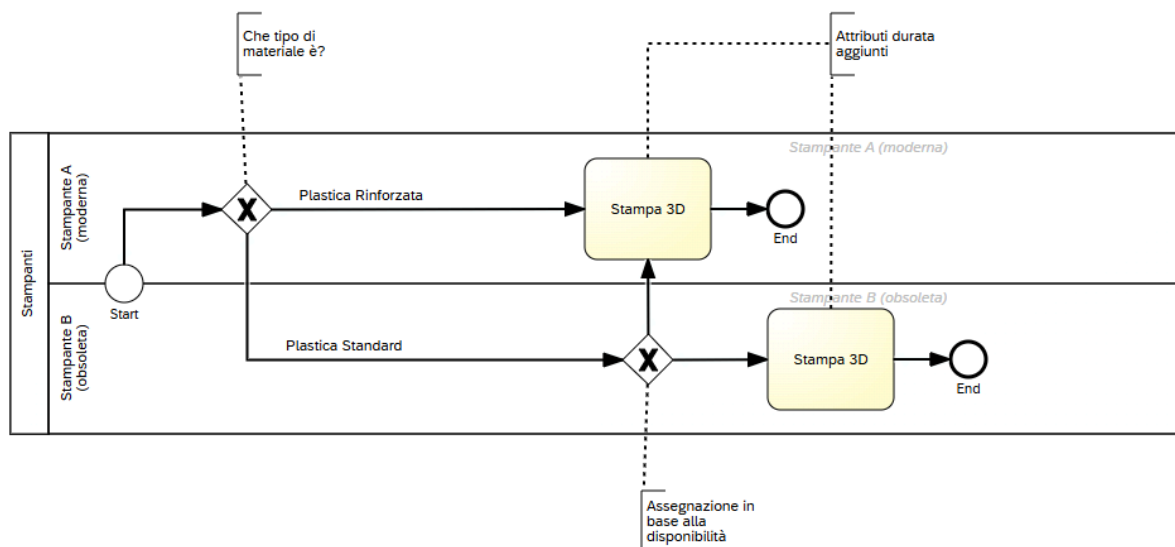


Fig.5: Scenario "Stampa 3D Componente" in Signavio

Criticità emerse:

- 1) **Falsa stima della Durata (Fallimento R9)**: In particolare nel secondo gateway, ci potrebbero essere dei problemi di *race conditions* o di *loop* infiniti, portando a stime errate o ad errori di simulazione.
- 2) **Incorretta allocazione delle risorse (Fallimento R8)**: L'arrivo di *token* indicanti "plastica rinforzata" in modo costante nel tempo potrebbe di fatto escludere la Stampante A dalla stampa di "plastica standard", eliminando la

classica gestione FIFO e rendendo le due stampanti esclusive sui materiali da loro gestiti (la stampante A stamperà solo con la plastica rinforzata e la stampante B solo con la plastica standard), rendendo la simulazione inefficiente e inesatta.

Modellazione *To-Be*

Replichiamo lo stesso scenario in ARMS:

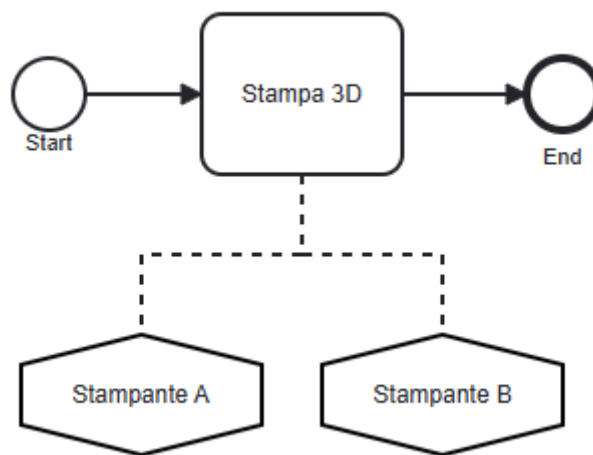


Fig.6: Scenario “Stampa 3D Componente” in ARMS

Risultato della Validazione

In ARMS il diagramma risulta più lineare e la logica è gestita in *background* tramite l'entità *Compatibility*, che agisce da ponte tra attività ed esecutore. Il simulatore infatti, in caso abbia a disposizione della plastica rinforzata, attiverà obbligatoriamente la Stampante A; invece in caso di disponibilità di plastica standard, interrogherà la matrice di compatibilità e farà avanzare il tempo in base alla risorsa libera in quel determinato momento (rispetto **R8**). Questo garantisce una stima del *throughput* realistica e sensibile allo stato delle risorse, senza dover disegnare flussi complessi (rispetto **R9**).

Lo si può notare in questa immagine, dove in una richiesta di 10 stampe 3D con plastica standard e 4 con plastica rinforzata, il simulatore ha indicato un consumo di tempo pari a 140 minuti e un'attività nel *task* Stampa 3D che denota come la Stampante A agisca sia sulle stampe con plastica rinforzata che su quelle con plastica standard, mentre la Stampante B agisce solo sulle seconde:

Data for Stampa 3D

Times of assigned executors



Stampante A
Stampante B

Products processed at assigned executors



Stampante A
Stampante B

Fig.7: Uso dei diversi esecutori nello scenario “Stampa 3D Componente”

5.3 Vincoli su Risorse Accessorie e Priorità (R5 + R7)

In questo scenario si simula un reparto di verniciatura dove arrivano due flussi di ordini: ordini urgenti e ordini standard (entrambi hanno una durata di 10 minuti). Ci sono due operatori disponibili ma una sola pistola a spruzzo (vincolo Tecnico: **R5**) e, nel caso in cui arrivino due ordini contemporaneamente (uno urgente e l'altro standard), la singola pistola a spruzzo deve essere assegnata all'ordine urgente (vincolo Gestionale: **R7**).

Modellazione As-Is

Possibile replica dello scenario:

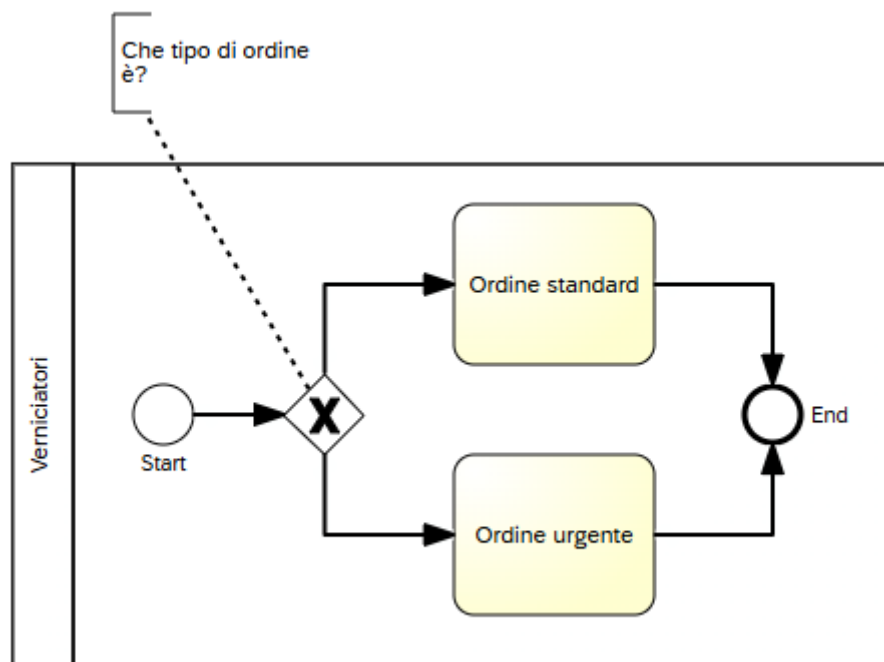


Fig.8: Scenario “Collo di Bottiglia in Verniciatura” in Signavio

Criticità emerse:

- 1) Cecità alle Risorse Ausiliarie (Fallimento R5):** BPMN standard non ha un concetto nativo per dire “Questo *task* richiede l’operatore e lo strumento X” in quanto non gestisce nativamente la doppia allocazione. La simulazione quindi avvierebbe la simulazione solamente in base al tipo di ordine ignorando della disponibilità della pistola.
- 2) Gestione FIFO (Fallimento R7):** I motori di simulazione standard gestiscono le code tipicamente in logica FIFO (*First In, First Out*). Se l’ordine standard arriva un secondo prima di quello urgente, la risorsa verrà allocata al primo. Manca infatti un attributo *priority* che consenta all’ordine urgente di acquisire precedenza nella coda di esecuzione.

Modellazione To-Be

Replichiamo questo scenario in ARMS:

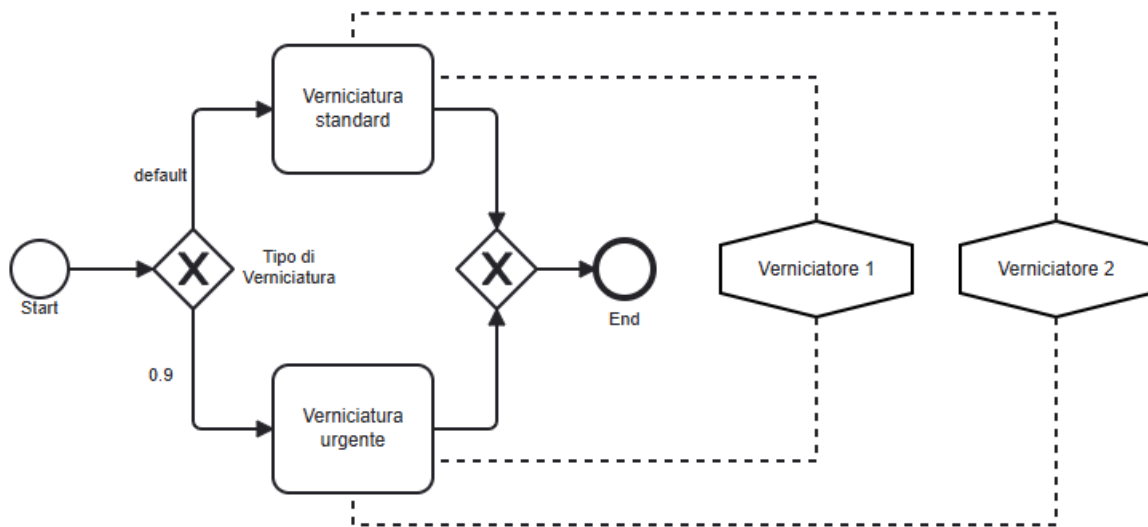


Fig.9: Scenario “Collo di Bottiglia in Verniciatura” in ARMS

Risultato della Validazione

Quando il simulatore processa la coda degli eventi:

- 1) Verifica la disponibilità dell'accessorio;
- 2) Rileva il conflitto causato dalla richiesta simultanea dell'unica risorsa disponibile;
- 3) Legge l'attributo *priority*;
- 4) Assegna l'accessorio all'ordine urgente, ponendo quello standard in stato di attesa fino al rilascio della risorsa (rispetto **R5** e **R7**).

Il simulatore infatti ha correttamente simulato che per 10 verniciature della durata di 10 minuti il tempo totale, data la disponibilità di una sola pistola a spruzzo, sarà di 100 minuti.

5.4 Attività in Batch e Accumulo Dinamico (R2)

In questo scenario, simuliamo un processo in cui l'efficienza energetica e operativa impone che una macchina non lavori sul singolo pezzo, ma accumuli una certa quantità di materiale prima di avviarsi. Una “Lavatrice Industriale” che lava dei vestiti con una capacità e un requisito operativo preciso: deve trattare **50 pezzi contemporaneamente** (vincolo di *Batch*: **R2**). L'attività quindi non deve partire quando arriva il primo pezzo ma deve restare in attesa finché non avrà accumulato

le 50 unità di capi necessarie, che una volta arrivate devono essere lavorate in un unico ciclo di durata fissa.

Modellazione As-Is

Possibile replica dello scenario:

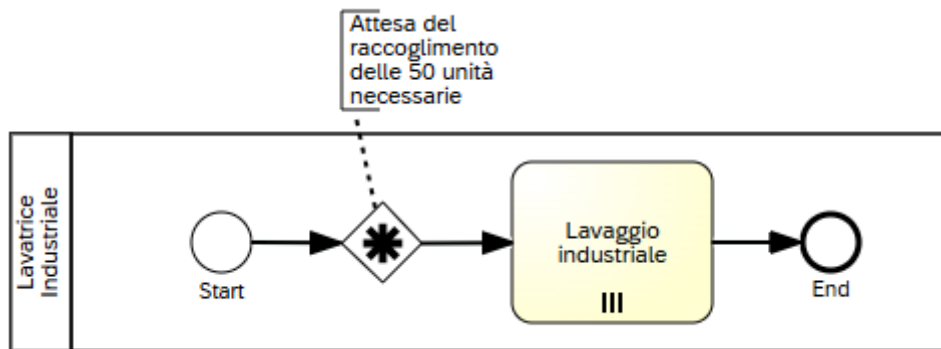


Fig.10: Scenario "Lavaggio in Lotti" in Signavio

Criticità emerse:

1) **Non è simulabile:** la *Complex Gateway* (che replica l'attesa nel raccogliere le 50 unità di vestiario) non è simulabile in Signavio. Se invece venisse messo in atto un *workaround* del modello, eliminando questo particolare tipo di *gateway* e rendendo il modello simulabile (un'ipotesi comunque irrealistica ma che si supponga sia vera), emergerebbero altri problemi:

- a) **Esecuzione Impaziente:** Signavio opera secondo una logica a singolo *token*, quindi, all'arrivo del primo capo, il motore rileva la risorsa libera e avvia il *task*;
- b) **Errore di Throughput e Costi:** A fronte di 50 capi in ingresso, Signavio simulerebbe 50 cicli distinti, non raggruppando tempo e costi in un ciclo (come previsto dall'attività in *batch*) e facendo produrre valori inesatti alla simulazione.

Modellazione To-Be

Replica di questo scenario in ARMS:

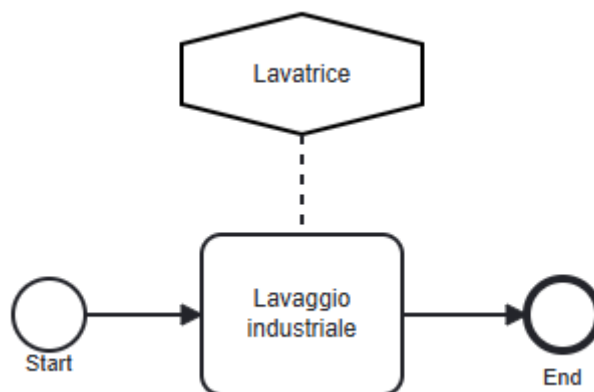


Fig.11: Scenario "Lavaggio in Lotti" in ARMS

Risultato della Validazione

In questo processo:

- 1) Arriva il primo pezzo: l'attività non parte (condizione non soddisfatta);
- 2) Arrivano i pezzi 2-49: l'attività resta comunque ferma;
- 3) Arriva il pezzo numero 50: la transizione si attiva e l'attività esegue **un solo ciclo** della durata prevista, con i risultati che riflettono correttamente l'utilizzo della risorsa (1 ciclo invece di 50) (rispetto **R2**).

Il simulatore ha correttamente interpretato l'esecuzione a lotti, in cui in un esempio di 200 vestiti, con la lavatrice che ne lava 50 alla volta con una durata di 30 minuti, ha eseguito 4 *loop* stimando una durata totale del processo di 120 minuti:

Simulation results

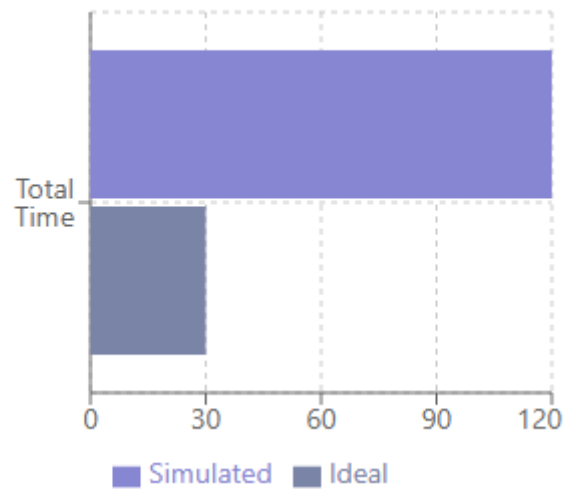


Fig.12: Tempo stimato nella simulazione dello scenario “Lavaggio in Lotti”

5.5 Ottimizzazione Operativa (R3)

In questo scenario, simuliamo un processo di revisione orologiera, dove la continuità del lavoro è essenziale per la qualità del lavoro svolto. In questo processo definiamo due attività, una di “Revisione del Movimento” e l’altra di “Collaudo Funzionale”, e si hanno a disposizione due operatori. Una volta revisionato e riassemblato il movimento però c’è la necessità di svolgere il collaudo, e questo deve essere eseguito dallo **stesso operatore** che ha eseguito la revisione in quanto conosce i dettagli del pezzo specifico (vincolo di Affinità: **R3**).

Modellazione As-Is

Replica dello scenario in Signavio:

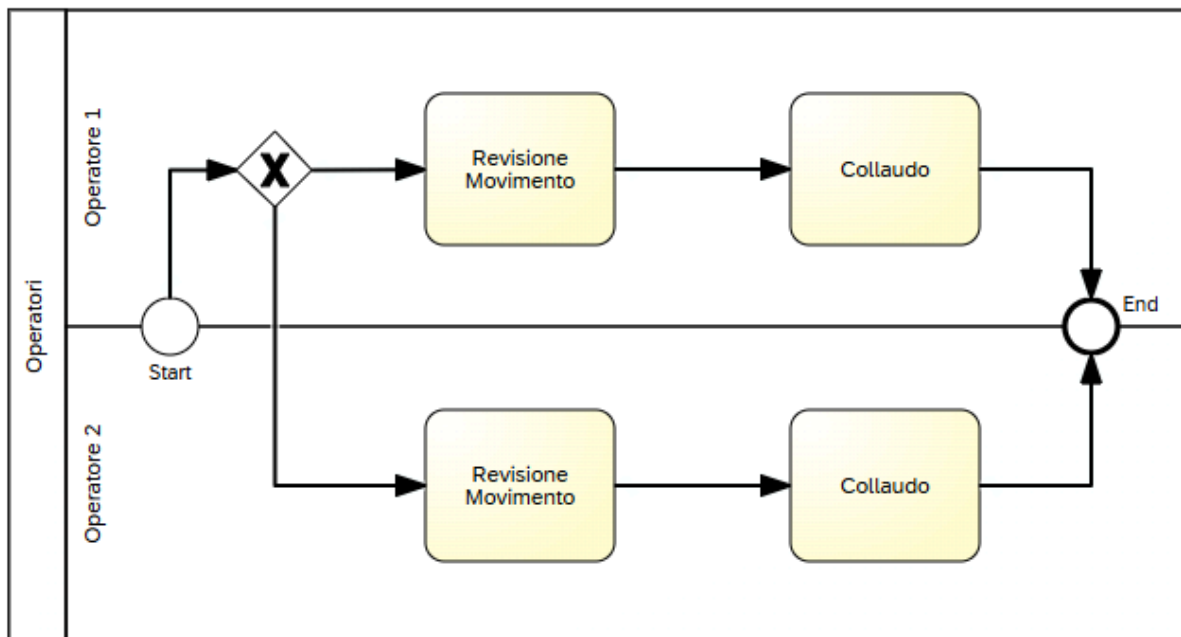


Fig.13: Scenario "Revisione Movimento" in Signavio

Criticità emerse:

- 1) **Allocazione errata (Fallimento parziale di R3):** sebbene **R3** appaia rispettato grazie all'uso di *Exclusive Gateway*, l'assegnazione del flusso (quindi in questo caso il movimento da revisionare) non avverrebbe in base alla disponibilità di manodopera ma in base ad un sistema di probabilità configurabile nei *Sequence Flow*, ciò potrebbe portare a simulazioni imprecise poiché, in un dato momento, un operatore potrebbe trovarsi a dover gestire due movimenti da revisionare, mentre l'altro tecnico non ne ha nessuno.

Modellazione To-Be

Replica di questo scenario in ARMS:

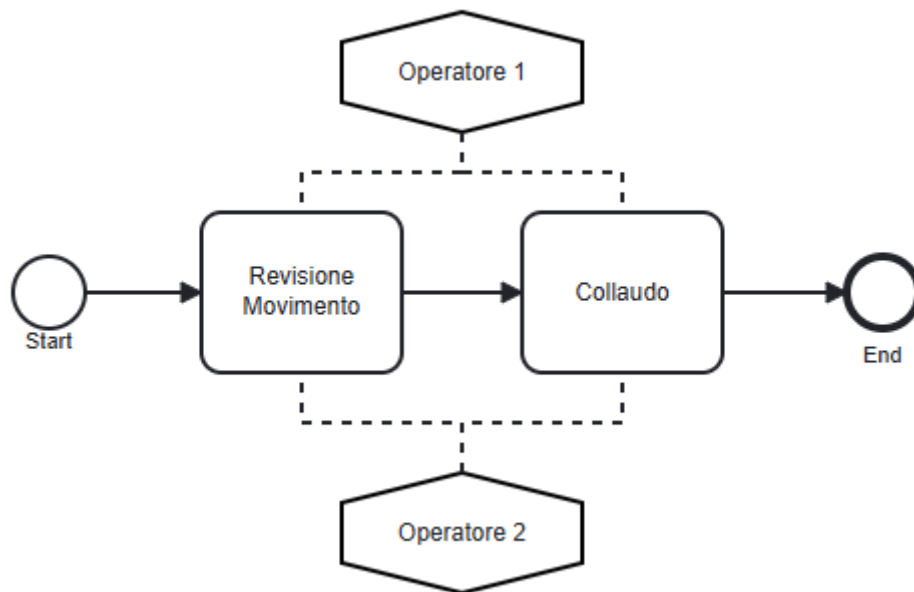


Fig.14: Scenario "Revisione Movimento" in ARMS

Risultato della Validazione

La soluzione in ARMS mostra un report dove ogni collaudo è eseguito dalla stessa persona che ha revisionato il movimento (rispetto **R3**) ma soprattutto ha la capacità di assegnare il movimento in base alla disponibilità di manodopera producendo una simulazione più fedele alla realtà. Il rispetto del requisito preso in considerazione lo si può evincere in queste immagini dove entrambi gli esecutori hanno impiegato lo stesso periodo di tempo in entrambi i *task* del processo:

Data for A1

Times of assigned executors



■ E1 ■ E2

Data for A2

Times of assigned executors



■ E1 ■ E2

Products processed at assigned executors



■ E1 ■ E2

Products processed at assigned executors



■ E1 ■ E2

Fig.15: Attività degli esecutori nello scenario “Revisione Movimento”

5.6 Caso Studio Integrato

Conclusa la validazione requisito per requisito, viene presentato un ipotetico scenario integrato che dimostra la congiunzione di tutti i requisiti in un unico flusso produttivo, con schermate che raffigurano tutti i tipi di dati ottenibili in una simulazione in ARMS. L'obiettivo è simulare la linea di assemblaggio di un'azienda produttrice di biciclette elettriche, declinata in due modelli: “Base” e “Performance”.

Il processo si articola in quattro fasi sequenziali:

- 1) **Approvvigionamento (R6 - Inventario):** Il processo attinge da tre magazzini distinti: “Telai Grezzi”, “Motori Elettrici” e “Batterie”. L'attività non può avviarsi se lo stock fisico è esaurito.
- 2) **Verniciatura a Lotti (R2 - Batch):** I telai grezzi entrano in un forno industriale in cui, per ottimizzare i costi energetici, vengono verniciati in lotti da 5.

3) Installazione Motore (R1 - Combinazione, R8 - Compatibilità e R9 -

Durata Variabile): Sono disponibili due esecutori del *task* di installazione del motore elettrico: un **robot automatizzato** (che impiega 5 minuti per il montaggio) e un **operatore umano** (che invece ne impiega 20 di minuti) per vincolare il processo rispetto a **R9**. Si ipotizza inoltre che il motore elettrico del modello “Performance” sia particolarmente difficoltoso da montare richiedendo l’intervento obbligatorio dell’operatore umano, al contrario il robot potrà concentrarsi sul montaggio dei motori sui modelli “Base” (vincolo **R8**). Per costruire un “Motore Installato” (dove si intende un’*e-bike* senza la batteria ma con telaio e motore già montati), si necessita di un telaio verniciato e di un motore elettrico (vincolo **R1**).

4) Assemblaggio Finale (R1, R5 - Accessori e R7 - Priorità): Per produrre una *e-bike* completa infine si utilizzano un “Motore Installato” e una batteria (vincolo **R1**), l’assemblaggio delle biciclette “Performance” hanno priorità maggiore rispetto a quelle “Base” (vincolo **R7**), questa operazione richiede un tablet di diagnostica per convalidare l’assemblaggio e questo reparto, composto da due assemblatori, ne possiede uno solo (vincolo **R5**), se quindi il tablet è in utilizzo, nessun’altra bici potrà essere assemblata nel mentre. Inoltre impostiamo la durata del *task* a 20 minuti per l’Assemblatore 1 e a 25 per l’Assemblatore 2, simulando una maggior maestria nella professione del primo esecutore.

Ecco come si presenta questo processo in ARMS:

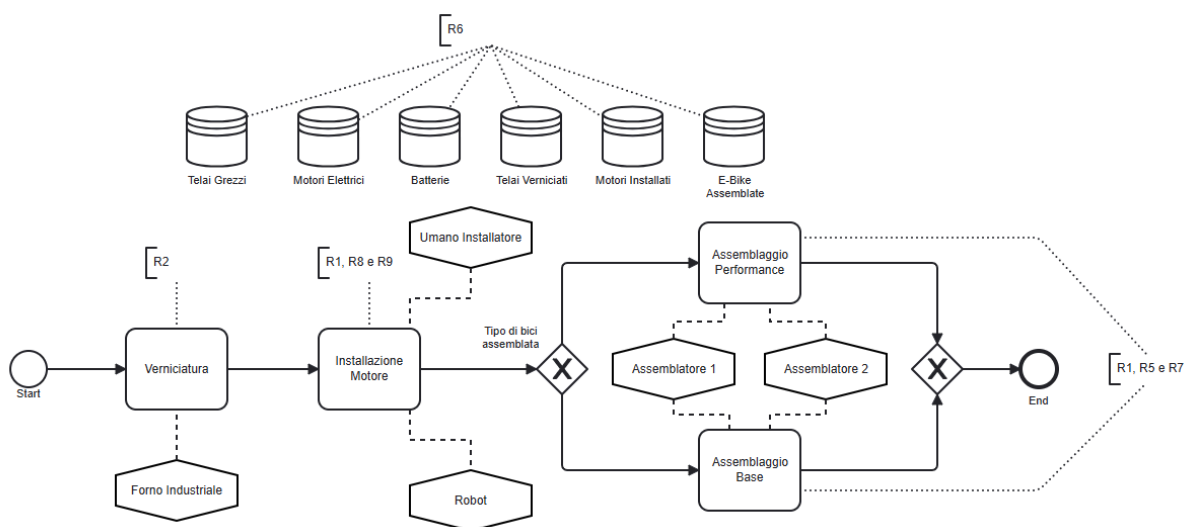


Fig.16: Scenario “Assemblaggio *e-bike*” in ARMS

Ecco alcuni risultati ottenuti nel caso in cui si volessero produrre 11 *e-bike* di tipo “Base” e 4 di tipo “Performance”:

Simulation results

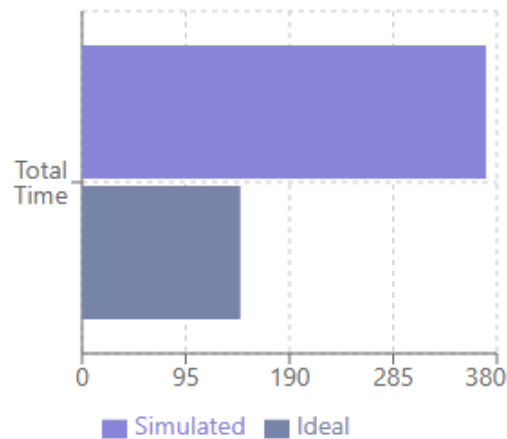


Fig.17: Tempo totale (370 minuti) ottenuto simulando il processo “Assemblaggio *e-bike*”

Data for Forno Industriale

Time-frame for Forno Industriale-1



■ Verniciatura

Times for Forno Industriale-1

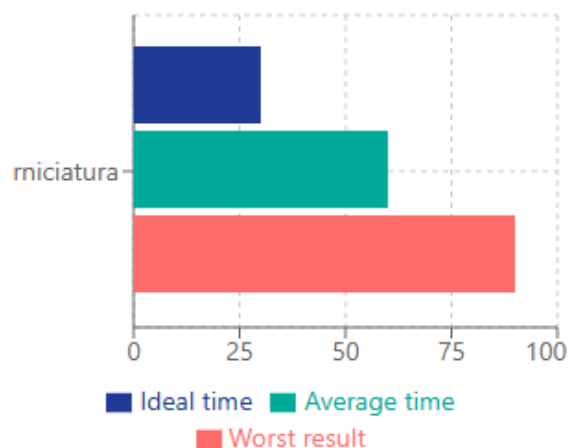
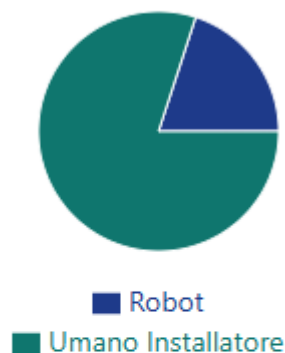


Fig. 18: Dati riguardanti l’attività dell’esecutore “Forno Industriale” che svolge il task “Verniciatura”

Il diagramma a torta espone il tempo impiegato nell'eseguire il *task* (90 minuti) e quello a barre invece i tempi di attesa (ideale, medio e peggiore) che i lotti hanno subito.

Data for Installazione Motore

Times of assigned executors



Products processed at assigned executors

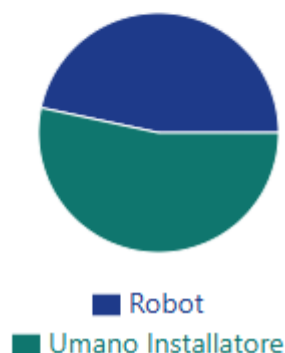


Fig. 19: Dati riguardanti all'attività "Installazione Motore"

Senza contare i tempi di attesa, il robot ha impiegato 35 minuti per installare 7 motori (solo sui modelli "base" per il vincolo a lui imposto), mentre l'operatore umano ne ha impiegati 140 per installarne 8.

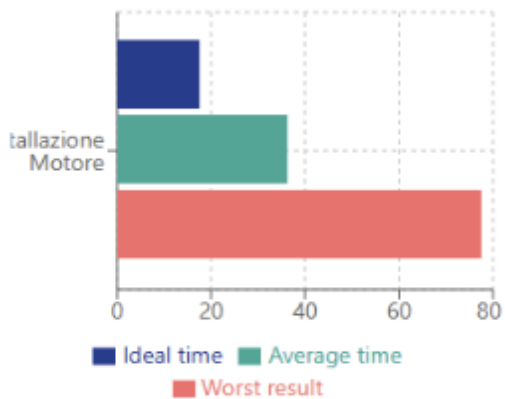
Data for Umano Installatore

Time-frame for Umano Installatore-1



■ Installazione Motore

Times for Umano Installatore-1



Data for Robot

Time-frame for Robot-1



■ Installazione Motore

Times for Robot-1

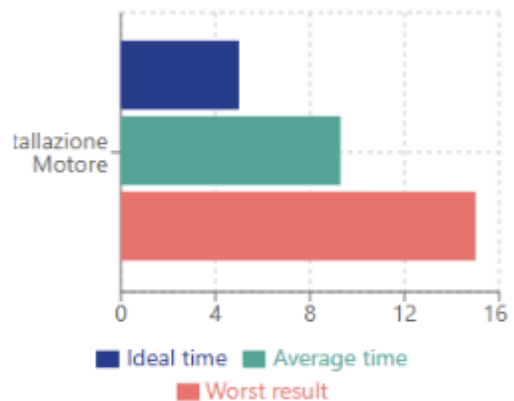


Fig. 20: Dati riguardanti l'attività degli esecutori "Umano Installatore" e "Robot "che svolgono il task "Installazione Motore"

Da questa schermata si nota che prima di eseguire un'installazione di un motore il tempo medio di attesa è stato di 36,25 minuti per l'operatore umano mentre per il robot si parla di circa 9,29 minuti.

Data for Assemblaggio Performance

Times of assigned executors



■ Assemblatore 1
■ Assemblatore 2

Products processed at assigned executors



■ Assemblatore 1
■ Assemblatore 2

Data for Assemblaggio Base

Times of assigned executors



■ Assemblatore 1
■ Assemblatore 2

Products processed at assigned executors



■ Assemblatore 1
■ Assemblatore 2

Fig. 21: Dati riguardanti i *task* "Assemblaggio Performance" e "Assemblaggio Base"

In questa simulazione, il *task* "Assemblaggio Performance" è stato attuato principalmente dall'Assemblatore 1 (60 minuti impiegati per 3 assemblaggi contro i 25 minuti per 1 assemblaggio effettuato dall'Assemblatore 2). L'"Assemblaggio Base" invece è stato ripartito in modo equilibrato (100 minuti impiegati per 5 assemblaggi da parte dell'Assemblatore 1 e 150 minuti impiegati per 6 assemblaggi dell'Assemblatore 2).

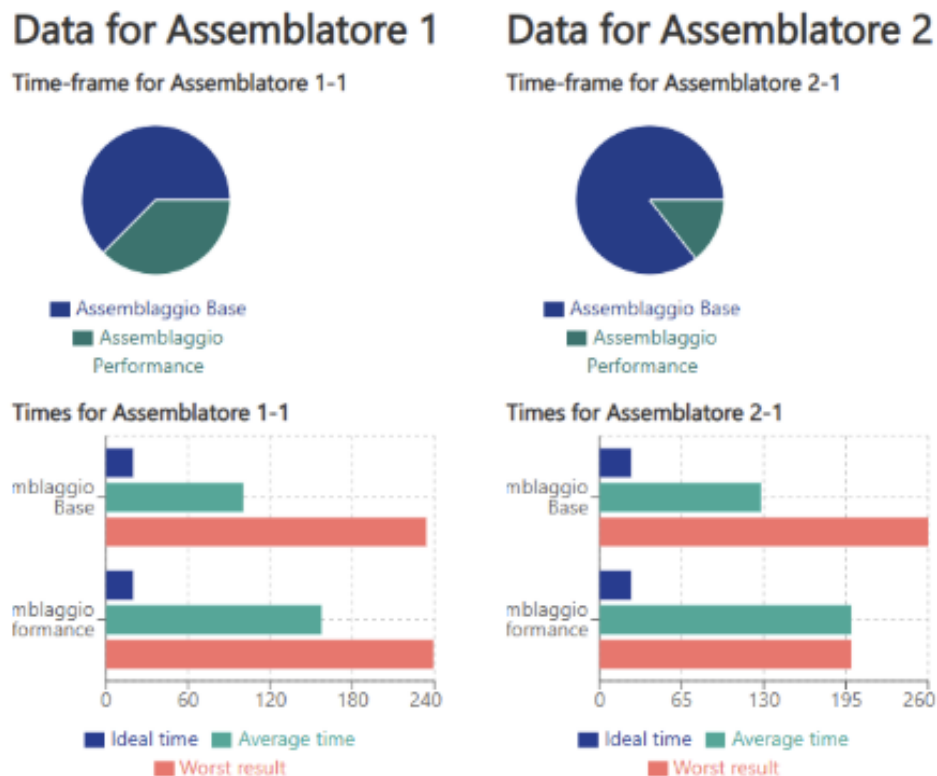


Fig. 22: Dati riguardanti l'attività degli esecutori "Umano Installatore" e "Robot "che svolgono il task "Installazione Motore"

In questa schermata invece vengono evidenziate un paio di cose (oltre ai tempi di esecuzione già esposti nell'immagine precedente):

- Per l'Assemblatore 1, il tempo medio di attesa in "Assemblaggio Base" è stato di 100,2 minuti mentre in "Assemblaggio Performance" l'attesa media è di 157,33 minuti circa.
- Per l'Assemblatore 2 invece il tempo medio di attesa in "Assemblaggio Base" si attesta sui 127,5 minuti mentre in "Assemblaggio Performance" l'attesa media è stata di 199 minuti.

Il tempo di attesa nell'"Assemblaggio Performance" maggiore rispetto a quello dell'"Assemblaggio Base", sebbene l'attributo *priority* sia presente e correttamente impostato in modo tale che le *e-bike* di tipo performance abbiano maggior priorità nella costruzione, accade perché i *token* di richiesta di assemblaggio delle *e-bike* vengono generati nell'ordine di dichiarazione. Il tempo di attesa superiore è dunque imputabile all'ordine di inserimento delle richieste: le *e-bike* "Base" sono state introdotte nel sistema prima di quelle "Performance". Invertendo l'ordine di richiesta, l'"Assemblaggio Performance" registrerebbe un tempo medio di attesa inferiore rispetto al modello "Base".

CONCLUSIONE

Il presente lavoro di tesi ha affrontato le criticità legate alla modellazione e simulazione dei processi produttivi nel settore manifatturiero, analizzando i limiti dello standard **BPMN** e validando l'efficacia di **ARMS**, sebbene questa si trovi ancora in fase di sviluppo, come soluzione per colmare il divario semantico esistente, illustrando inoltre la reportistica fornita dall'applicazione.

Dall'analisi della letteratura e dello stato dell'arte è emerso come la natura *activity-centric* della notazione standard si è rivelata inadeguata nel rappresentare le dinamiche fondamentali dei processi produttivi in ambito manifatturiero. Tali mancanze, classificate nei nove requisiti fondamentali (**R1-R9**) conducono molto spesso a simulazioni "ottimistiche", a dati inattendibili o alla non possibilità di simulare il processo in quanto mancante di *features* importanti, compromettendo la capacità decisionale in ottica di Industria 4.0.

L'adozione di un approccio *resource-centric* ha permesso di superare questi ostacoli, integrando la visione fisica all'interno del flusso logico del processo, senza rinunciare alla comprensibilità grafica del BPMN.

La validazione sperimentale ha dimostrato la superiorità di ARMS in:

- **Gestione dei Materiali (R1, R6);**
- **Eterogeneità delle Risorse (R8, R9);**
- **Logiche di Priorità e Batch (R2, R5, R7);**
- **Gestione Operativa (R3).**

Tuttavia, il lavoro ha anche evidenziato come ARMS sia un progetto ancora in fase di evoluzione. Come discusso nella metodologia l'attuale implementazione non supporta ancora la totalità dei costrutti BPMN (*Pools, Groups* e *Sub-processes*), ed inoltre presenta alcuni problemi in fase di modellazione e simulazione, ad esempio non fornisce *feedback* sul dove si trovano gli errori che fanno fallire la simulazione, funzionalità che saranno presumibilmente implementate negli sviluppi futuri.

Nonostante ciò, ARMS presenta un ottimo potenziale per diventare un buon strumento di simulazione in ambito manifatturiero per coloro che hanno necessità di simulare dei processi produttivi in ambito manifatturiero ma non hanno maestria con programmi di simulazione a eventi discreti e vogliono mantenere la semplicità grafica di BPMN.

BIBLIOGRAFIA

- [1] C. Stryker e I. Belcic, "Che cos'è il BPMN (Business Process Modeling and Notation)?" IBM. [Online]. Disponibile su: <https://www.ibm.com/it-it/think/topics/bpmn>. [Consultato il: 10 ott. 2025].
- [2] G. Aagesen e J. Krogstie, "BPMN 2.0 for Modeling Business Processes," in *Handbook on Business Process Management 1*, Berlino, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 219–250. [Online]. Disponibile su: https://doi.org/10.1007/978-3-642-45100-3_10. [Consultato il: 14 ott. 2025].
- [3] ProcessMind, "BPMN Diagram Construction Guide," *ProcessMind Resources*. [Online]. Disponibile su: <https://processmind.com/resources/docs/bpmn-2-0/constructing-a-bpmn-diagram>. [Consultato il: 28 nov. 2025].
- [4] A. García-Domínguez, M. Marcos e I. Medina, "A comparison of BPMN 2.0 with other notations for manufacturing processes," in *AIP Conference Proceedings*, vol. 1431, n. 1, 2012, pp. 593–600. [Online]. Disponibile su: <https://doi.org/10.1063/1.4707613> [Consultato il: 15 ott. 2025].
- [5] H. Leopold, J. Mendling e O. Günther, "Learning from Quality Issues of BPMN Models from Industry," *IEEE Software*, vol. 33, n. 4, pp. 26–33, lug.-ago. 2016. [Online]. Disponibile su: <https://doi.org/10.1109/ms.2015.81> [Consultato il: 18 ott. 2025].
- [6] K. Rosenthal, B. Ternes e S. Strecker, "Business Process Simulation: A Systematic Literature Review," in *Proceedings of the 26th European Conference on Information Systems (ECIS)*, Portsmouth, UK, 23-28 giu. 2018, Art. n. 199. [Online]. Disponibile su: https://aisel.aisnet.org/ecis2018_rp/199 [Consultato il: 20 ott. 2025].
- [7] M. R. Khabbazi, M. K. Hasan, R. Sulaiman e A. Shapi'i, "Business Process Modelling in Production Logistics: Complementary Use of BPMN and UML", *Middle-East Journal of Scientific Research*, vol. 15, n. 4, pp. 516–529, 2013. [Online]. Disponibile su: <https://api.semanticscholar.org/CorpusID:22160383> [Consultato il: 28 ott. 2025].

- [8] Y. Kirikkayis, M. Winter e M. Reichert, “A User Study on Modeling IoT-Aware Processes with BPMN 2.0,” *Information*, vol. 15, n. 4, art. n. 229, 2024. [Online]. Disponibile su: <https://doi.org/10.3390/info15040229> [Consultato il: 29 ott. 2025].
- [9] I. Pihir, K. Tomičić-Pupek e N. Vrček, “Challenges of Processes Simulation with Dynamic Batch Processing Activities,” *Croatian Operational Research Review*, vol. 9, n. 1, pp. 99–113, 2018. [Online]. Disponibile su: <https://doi.org/10.17535/corr.2018.0009> [Consultato il: 3 nov. 2025].
- [10] B. Olde Hampsink, “Assessing the validity of the simulation of manufacturing processes in BPM tools,” Tesi di Laurea Magistrale, School of Industrial Engineering, Eindhoven Univ. of Technology, Eindhoven, Paesi Bassi, 2018. [Online]. Disponibile su: <https://research.tue.nl/en/studentTheses/9daf109d-b9fd-4380-8ab9-7b79ec75e9d9> [Consultato il: 5 nov. 2025]
- [11] A. Di Iorio, M. Ferrati e D. Rossi, “A Conceptual Framework for the Simulation of Manufacturing Process,” manoscritto non pubblicato.
- [12] A. Di Iorio, M. Ferrati e D. Rossi, “ARMS: Activity-Resource Modelling Simulator,” in *Simulation Tools and Techniques*, A. A. Juan, J. L. Guisado-Lizar, M. J. Morón-Fernández e E. Perez-Bernabeu, Cur., ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 603. Cham: Springer, 2025, pp. 167–185. [Online]. Disponibile su: https://doi.org/10.1007/978-3-031-87345-4_12.
- [13] S. Zor, D. Schumm e F. Leymann, “A Proposal of BPMN Extensions for the Manufacturing Domain,” in *Proceedings of the 44th CIRP International Conference on Manufacturing Systems*, Madison, WI, USA, 1-3 giu. 2011. [Online]. Disponibile su: <https://www.iaas.uni-stuttgart.de/publications/INPROC-2011-18-A-Proposal-for-BPM-N-Extensions-for-the-Manufacturing-Domain.pdf> [Consultato il: 24 nov. 2025]
- [14] M. Aspidou, “Extending BPMN for modeling manufacturing processes,” Tesi di Laurea Magistrale, Eindhoven Univ. of Technology, Eindhoven, Paesi Bassi, 2017. [Online]. Disponibile su: <https://research.tue.nl/en/studentTheses/7a6913ff-13bc-47b7-9881-e58636f610d1>. [Consultato il: 24 nov. 2025]

[15] V. H. M. Ribeiro, J. Barata e P. R. da Cunha, "A BPMN Extension to Model Inter-Organizational Processes in Industry 4.0," in *Information Systems Development: Crossing Boundaries between Development and Operations (DevOps) in Information Systems (ISD2021 Proceedings)*, E. Insfran et al., Cur. Valencia, Spagna: Universitat Politècnica de València, 8-10 set. 2021. [Online]. Disponibile su: <https://aisel.aisnet.org/isd2014/proceedings2021/currenttopics/1/>. [Consultato il: 25 nov. 2025]

[16] R. Petrasch, "I4PML - Industry 4.0 Process Modeling Language for Model Driven Development," Beuth Hochschule für Technik Berlin, Berlino, Germania, Rapp. Tecn. TR20170001b, feb. 2017. [Online]. Disponibile su: https://www.researchgate.net/profile/Roland-Petrasch/publication/313599337_I4PML-Industry_40_Process_Modeling_Language_for_Model_Driven_Development_-_Definition_of_I4PML_Profile_for_UML_Activity_Diagrams_in_conjunction_with_Enterprise_Integration_Patterns/links/61b421261d88475981df36d7/I4PML-Industry-40-Process-Modeling-Language-for-Model-Driven-Development-Definition-of-I4PML-Profile-for-UML-Activity-Diagrams-in-conjunction-with-Enterprise-Integration-Patterns.pdf [Consultato il: 25 nov. 2025].

[17] Y. Kirikkayis, F. Gallik, M. Winter e M. Reichert, "BPMNE4IoT: A Framework for Modeling, Executing and Monitoring IoT-Driven Processes," *Future Internet*, vol. 15, n. 3, art. n. 90, feb. 2023. [Online]. Disponibile su: <https://doi.org/10.3390/fi15030090> [Consultato il: 26 nov. 2025].

RINGRAZIAMENTI

Un caloroso ringraziamento va al Dottorando Marco Ferrati per il prezioso supporto fornitomi durante la fase di sviluppo e *debugging* dei modelli dei vari scenari di processi produttivi nell'ambiente ARMS. Grazie alla sua disponibilità è stato possibile validare correttamente tutti i requisiti all'interno della fase di implementazione sperimentale, apportando un valore aggiunto al lavoro della tesi.