



Università della Calabria

Dipartimento di Economia, Statistica e Finanza “*Giovanni Anania*”

Corso di Laurea in Data Science
per le Strategie Aziendali

Sistemi Informativi: Processi di Business

LRL Tennis SRL

Studenti

Langè Francesco Giovanni 259558

Lindia Pierfrancesco 256641

Rausa Alessandro 252172

La seguente applicazione ha l'obiettivo di descrivere i processi di business relativi alla LRL Tennis SRL, un'azienda di produzione che si occupa di fornire attrezzatura tecnica per il tennis.

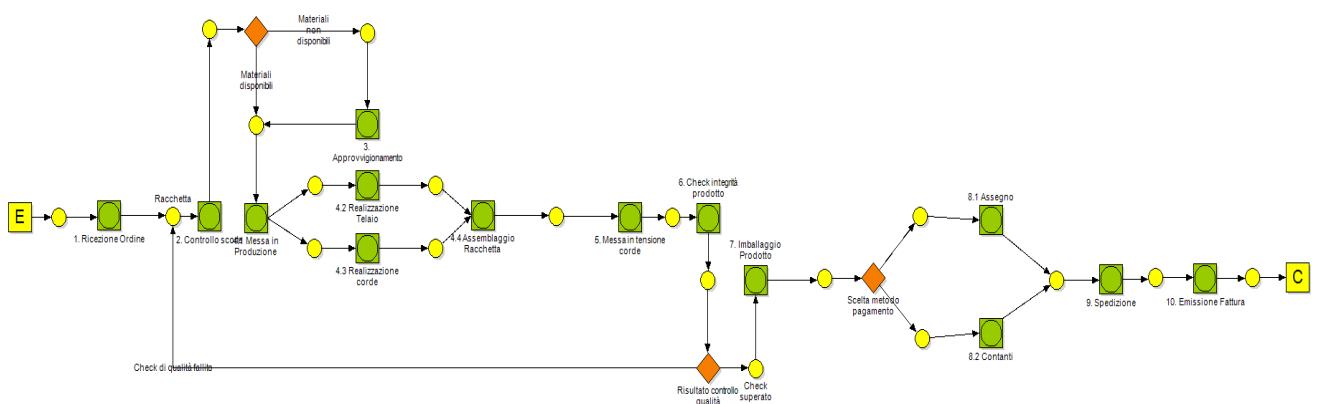
Nello specifico l'azienda è specializzata nella realizzazione di racchette da tennis (dal trattamento dei materiali fino all'assemblaggio), per poi occuparsi dell'imballaggio e della spedizione.

Attraverso il software Yasper si vuole rappresentare il processo di business, effettuarne l'analisi temporale e la simulazione. Attraverso il software YAWL invece simuliamo la parte gestionale.

RAPPRESENTAZIONE PROCESSO

Come accennato in precedenza, utilizziamo il software Yasper per la modellazione, rappresentazione e una simulazione del workflow.

Il processo modellato consta di 14 transizioni (task) e vari costrutti, tra cui lo XOR, l'AND e il ciclo.



- Il primo task riguarda la ricezione dell'ordine da parte dell'azienda.
- Successivamente si passa al Controllo delle scorte, da qui viene implementato il primo XOR Split relativo alla disponibilità dei materiali per la realizzazione del prodotto. In caso di indisponibilità dei materiali si procede all'Approvvigionamento (3.) e poi alla Messa in Produzione (4.1), in caso di disponibilità invece si procede direttamente a quest'ultimo task senza passare per il ramo alternativo.
- Una volta arrivati al Task di messa in produzione, si procede alla realizzazione vera e propria del prodotto. In parallelo vengono realizzati i due componenti principali della racchetta, cioè il Telaio (4.2) e le corde (4.3). I due rami paralleli vengono modellati attraverso l'AND Split e poi vengono riuniti attraverso l'AND Join in posizione 4.4, che corrisponde all'assemblaggio della racchetta.
- Una volta montate le corde sulla racchetta, esse vengono messe in tensione in modo tale da essere pronte per l'uso.
- A questo punto viene effettuato un controllo di qualità del prodotto (modellato attraverso un ciclo). Nel caso di check non superato si ritorna al place che precede il Controllo Scorte (2.) in modo da ripetere tutto il processo produttivo. In caso di check superato invece si procede all'Imballaggio del prodotto (7.).
- Superato il check e l'imballaggio è stato modellata la scelta del metodo di pagamento attraverso uno XOR Split. L'azienda dà la possibilità ai clienti di pagare attraverso Assegno (8.1) o Contanti (8.2).
- Una volta effettuato il pagamento si procede alla Spedizione e all'Emissione della fattura (10.).
- Completata quest'ultima attività il processo si considera concluso.

Parametri

Di seguito sono riportati i parametri utilizzati per ciascun task, con la specifica che l'unità di misura è espressa in minuti. Inoltre, si è utilizzato un tempo di interarrivo tra un ordine e l'altro di 20 minuti (specificato sull'Emittor).

Numero Task	Nome Task	Ruolo assegnato	Tempo medio esecuzione (e varianza)	Costo fisso	Costo variabile
1.	Ricezione Ordine	Commesso	10	1	1
2.	Controllo scorte	Magazziniere 1	5	1	1
3.	Approvvigionamento	Capo magazziniere	15	1	1
4.1	Messa in produzione	Magazziniere 2	5	1	1
4.2	Realizzazione Telaio	Operaio 1; Operaio 5	16	1	1
4.3	Realizzazione Corde	Operaio 2	10	1	1
4.4	Assemblaggio Racchetta	Operaio 3; Operaio 6	16	1	1
5.	Messa in tensione corde	Operaio 4	5	1	1
6.	Check integrità prodotto	Responsabile controllo qualità	13	1	1
7.	Imballaggio prodotto	Addetto spedizione 1	3	1	1
8.1	Assegno	Contabile 1	3	1	1
8.2	Contanti	Contabile 2	3	1	1
9.	Spedizione	Addetto spedizione 2	5	1	1
10.	Emissione fattura	Capo contabile	8	1	1

Probabilità scelte

Controllo scorte	
Materiali disponibili	0.95
Materiali non disponibili	0.05

Risultato controllo qualità	
Check superato	0.95
Check non superato	0.05

Scelta metodo di pagamento	
Assegno	0.50
Contanti	0.50

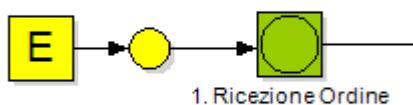
Analisi delle prestazioni

Riportiamo i valori del tempo di interarrivo, espressi in minuti, utilizzati per simulare i processi.

EMITTER	
1/LAMBDA	LAMBDA
10	0,1
15	0,066667
20	0,05
25	0,04
30	0,033333
45	0,022222
60	0,016667
75	0,013333
90	0,011111
100	0,01

Seguiranno degli esempi di diversi costrutti con relative analisi delle prestazioni.

Costrutto sequenziale



Il task 1. fa riferimento alla ricezione dell'ordine e viene gestito dall'operatore Commesso.

TASK RICEZIONE ORDINE									
1/lambda	1/mu	lambda	mu	rho	L	W	Lq	Wq	
10	10	0,1	0,1	1	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	
15	10	0,066667	0,1	0,666667	2	30	1,333333	20	
20	10	0,05	0,1	0,5	1	20	0,5	10	
25	10	0,04	0,1	0,4	0,666667	16,666667	0,266667	6,666667	
30	10	0,033333	0,1	0,333333	0,5	15	0,166667	5	
45	10	0,022222	0,1	0,222222	0,285714	12,85714	0,063492	2,857143	
60	10	0,016667	0,1	0,166667	0,2	12	0,033333	2	
75	10	0,013333	0,1	0,133333	0,153846	11,53846	0,020513	1,538462	
90	10	0,011111	0,1	0,111111	0,125	11,25	0,013889	1,25	
100	10	0,01	0,1	0,1	0,111111	11,11111	0,011111	1,111111	

Nella tabella precedente sono descritte le varie misure relative alla coda del task:

- *Tempo interarrivo*($1/\lambda$): Tempo medio di interarrivo tra un ordine e l'altro (*minuti*).
- *Tempo di Servizio* ($1/\mu$): Tempo medio di esecuzione del Task (*minuti*).
- *Tasso di interarrivo* (λ): Frequenza con cui gli ordini arrivano nel sistema.
- *Tasso di servizio* (μ): Frequenza di esecuzione del Task.
- *Rho o Tasso di utilizzo* (ρ): Il fattore di utilizzo del sistema, determinato dal rapporto tra il tempo di servizio e il tempo di interarrivo. Un valore di $\rho < 1$, indica che il sistema è stabile e che la coda

raggiungerà uno stato di stabilità, mentre un valore di ρ uguale o maggiore di 1 indica un sistema instabile.

- *Numero di Clienti nel Sistema (L)*: Il numero medio di clienti presenti nel sistema durante il tempo di osservazione.
- *Work Time (W)*: Il tempo medio di attraversamento del task (*minuti*).
- *Numero clienti in coda (Lq)*: Numero medio di clienti presenti nella coda del sistema durante il tempo di osservazione.
- *Tempo Attesa in Coda (Wq)*: Il tempo medio di attesa in coda (*minuti*).

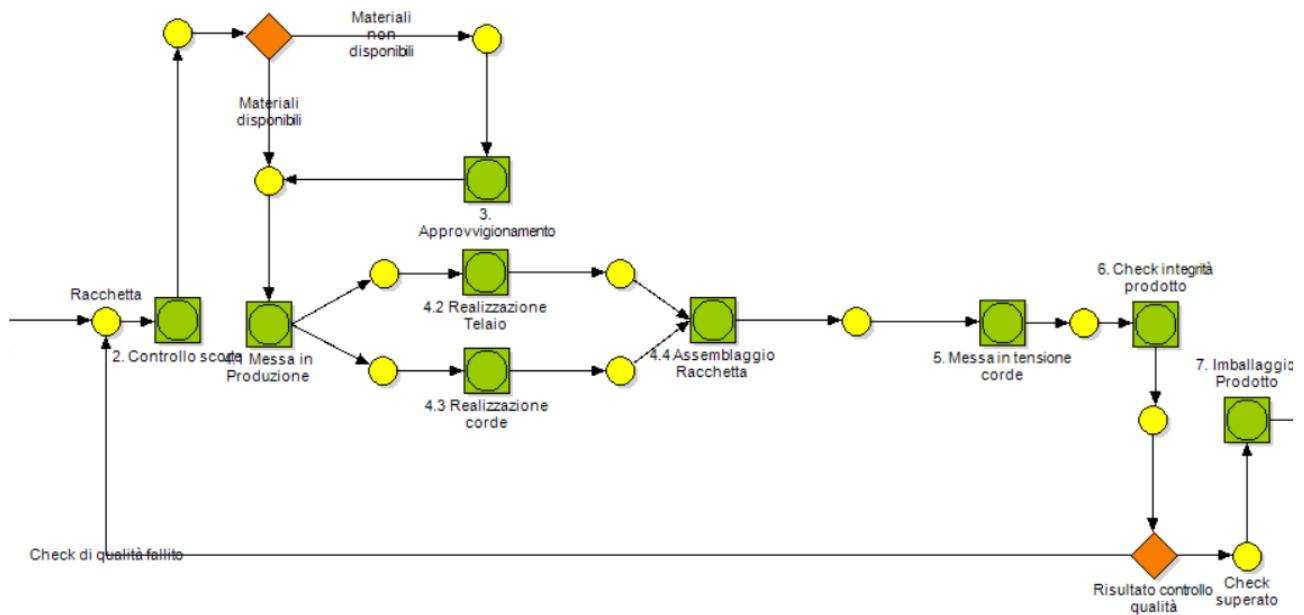
Un tempo di interarrivo più basso (ad esempio, 10 minuti) implica un arrivo più frequente degli ordini, mentre un tempo di interarrivo più alto (ad esempio, 100 minuti) implica un arrivo meno frequente.

In giallo sono segnalati i valori delle misure relative della coda del Task, in riferimento al tempo interarrivo utilizzato nella modellazione del processo in Yasper.

Per garantire la stabilità del sistema (*steady*), è essenziale mantenere il $\rho < 1$, poiché un $\rho = 1$ rende la coda instabile e quindi potrebbe non essere mai raggiunto lo stato di stabilità del sistema, mentre con un $\rho > 1$ lo stato di stabilità non verrà mai raggiunto e la coda crescerà all'infinito.

Ciò può essere ottenuto aumentando il numero di serventi del Task, oppure, regolando il *tasso di servizio* ($1/\mu$) ad esempio acquistando un software o un macchinario che permette di diminuire il tempo di servizio del Task.

Costrutto iterativo (Ciclo)



Superato il Task Check integrità del prodotto (6.) attraverso uno XOR Split si apre il ramo di backward del ciclo che, in caso di check di qualità non superato, fa tornare il processo al task di controllo scorte (2.) per poi iniziare nuovamente la fase di produzione.

Di seguito abbiamo riportato l'analisi delle prestazioni relative esclusivamente al ciclo.

1/LAMBDA	W Forward	P.ciclo	W Backward	W totale ciclo
10	-92,551959	0,05	-4,62759794	-97,17955683
15	157,26034	0,05	7,863016812	165,123353
20	112,18869	0,05	5,609434378	117,7981219
25	93,037736	0,05	4,651886785	97,68962249
30	84,843358	0,05	4,242167908	89,08552606
45	75,341978	0,05	3,767098914	79,1090772
60	71,856158	0,05	3,592807917	75,44896625
75	70,061032	0,05	3,503051603	73,56408367
90	68,971213	0,05	3,448560663	72,41977392
100	68,455961	0,05	3,422798043	71,87875891

Come si evince dal grafico, il ramo di backward e quello di forward coincidono. L'unica differenza è nel calcolo del worktime: infatti per calcolare il worktime del ramo B bisogna moltiplicare il W del ramo Forward per la probabilità di entrare nel ciclo (0.05). Mentre, il calcolo dell'analisi prestazionale totale del ciclo è data dalla somma tra il ramo di forward e il ramo di backward.

Questa tabella di analisi delle misure relative al ciclo fa riferimento all'utilizzo di code MM2 nei Task *realizzazione telaio* (4.2) e *aassemblaggio racchetta* (4.4), poiché da analisi condotte nella realizzazione del progetto si è riscontrata la formazione di colli di bottiglia in corrispondenza dei Task precedentemente citati(4.2, 4.4)

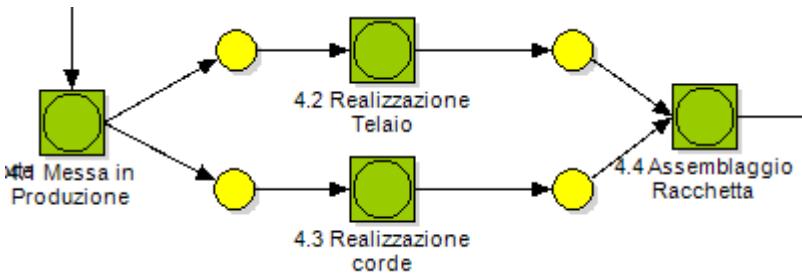
La seguente tabella fa, invece, riferimento all'analisi prestazionale del ciclo con l'utilizzo di code MM1 nei Task 4.2 e 4.4.

ANALISI CICLO MM1				
1/LAMBDA	W Forward	P.ciclo	W Backward	W totale ciclo
10	-218,75207	0,05	-10,9376037	-229,6896786
15	-100,09076	0,05	-5,00453784	-105,0952947
20	268,61114	0,05	13,43055724	282,0417021
25	151,05952	0,05	7,552976046	158,612497
30	120,39904	0,05	6,019951846	126,4189888
45	91,982623	0,05	4,59913114	96,58175393
60	82,810144	0,05	4,140507212	86,95065144
75	78,252138	0,05	3,912606884	82,16474457
90	75,522109	0,05	3,776105448	79,2982144
100	74,237993	0,05	3,711899646	77,94989256

Dall'analisi dei dati, emerge chiaramente che l'utilizzo di code MM1 nei task di realizzazione del telaio (4.2) e assemblaggio della racchetta (4.4) non è ottimale, specialmente quando il tempo di interarrivo è inferiore a 90 minuti. In queste condizioni, il sistema diventa congestionato, causando un aumento significativo del Work Time totale del ciclo. Questo aumento è attribuibile a un maggiore tasso di utilizzo (ρ) del servente, che porta a tempi di attesa più lunghi per i clienti oppure perché all'aumentare *del tempo di interarrivo ($1/\lambda$)*, rimanendo costante il *tempo di servizio ($1/\mu$)*, il *tasso di utilizzo (ρ)* del servente diminuisce.

L'implementazione di code MM2 si è dimostrata più efficiente, poiché l'aggiunta di un secondo servente ha mitigato i colli di bottiglia, riducendo il tempo di attesa e migliorando il flusso complessivo del processo. Questo ha portato a una diminuzione significativa del Work Time totale del ciclo, rendendo il sistema più stabile e capace di gestire un maggior numero di ordini senza incorrere in inefficienze.

Costrutto parallelo (AND Split)



All'interno del ciclo è presente il costrutto parallelo che consente l'esecuzione contemporanea di più task paralleli (dal task 4.1 al task 4.4). Questo costrutto è utilizzato per rappresentare situazioni in cui un processo può essere suddiviso in più rami paralleli che vengono eseguiti simultaneamente e indipendentemente l'uno dall'altro, per poi essere riuniti in un punto successivo.

Dopo la messa in produzione (4.1) infatti vengono avviati due task paralleli (4.2 e 4.3) per poi riunirli nel task (4.4).

L'osservazione da fare in questo caso è relativa ai task 4.2 e 4.4, infatti per queste due transizioni sono stati considerati due operatori a testa (e quindi una coda di tipo MM2), operazione resasi necessaria in seguito all'individuazione di colli di bottiglia

L'utilizzo di due operatori nei task di realizzazione del telaio e assemblaggio della racchetta riduce significativamente i tempi di attesa e i colli di bottiglia. Monitorare costantemente i parametri di interarrivo e di servizio è essenziale per mantenere il sistema in uno stato stabile, soprattutto per tempi di interarrivo bassi.

TASK REALIZZAZIONE CORDE									
1/lambda	1/mu	lambda	mu	rho	L	W	Lq	Wq	
9,5	10	0,105263	0,1	1,052632	-20	-190	-21,0526	-200	
14,25	10	0,070175	0,1	0,701754	2,352941	33,52941	1,651187	23,52941	
19	10	0,052632	0,1	0,526316	1,111111	21,11111	0,584795	11,11111	
23,75	10	0,042105	0,1	0,421053	0,727273	17,27273	0,30622	7,272727	
28,5	10	0,035088	0,1	0,350877	0,540541	15,40541	0,189663	5,405405	
42,75	10	0,023392	0,1	0,233918	0,305344	13,05344	0,071425	3,053435	
57	10	0,017544	0,1	0,175439	0,212766	12,12766	0,037327	2,12766	
71,25	10	0,014035	0,1	0,140351	0,163265	11,63265	0,022914	1,632653	
85,5	10	0,011696	0,1	0,116959	0,13245	11,3245	0,015491	1,324503	
95	10	0,010526	0,1	0,105263	0,117647	11,17647	0,012384	1,176471	

TASK REALIZZAZIONE TELAIO MM2									
1/lambda	1/mu	lambda	mu	rho	L	W	Lq	Wq	
9,5	16	0,105263	0,0625	0,842105	5,333333	55,00952	4,491228	42,66667	
14,25	16	0,070175	0,0625	0,561404	1,28	23,3636	0,718596	10,24	
19	16	0,052632	0,0625	0,421053	0,727273	19,44781	0,30622	5,818182	
23,75	16	0,042105	0,0625	0,336842	0,507937	18,04774	0,171094	4,063492	
28,5	16	0,035088	0,0625	0,280702	0,390244	17,36853	0,109542	3,121951	
42,75	16	0,023392	0,0625	0,187135	0,230216	16,58064	0,043081	1,841727	
57	16	0,017544	0,0625	0,140351	0,163265	16,32151	0,022914	1,306122	
71,25	16	0,014035	0,0625	0,112281	0,126482	16,20429	0,014202	1,011858	
85,5	16	0,011696	0,0625	0,093567	0,103226	16,14131	0,009659	0,825806	
95	16	0,010526	0,0625	0,084211	0,091954	16,11427	0,007743	0,735632	

TASK ASSEMBLAGGIO RACCHETTA MM2									
1/lambda	1/mu	lambda	mu	rho	L	W	Lq	Wq	
16	16	0,0625	0,0625	0,5	1	21,33333	0,5	8	
16	16	0,0625	0,0625	0,5	1	21,33333	0,5	8	
19	16	0,052632	0,0625	0,421053	0,727273	19,44781	0,30622	5,818182	
23,75	16	0,042105	0,0625	0,336842	0,507937	18,04774	0,171094	4,063492	
28,5	16	0,035088	0,0625	0,280702	0,390244	17,36853	0,109542	3,121951	
42,75	16	0,023392	0,0625	0,187135	0,230216	16,58064	0,043081	1,841727	
57	16	0,017544	0,0625	0,140351	0,163265	16,32151	0,022914	1,306122	
71,25	16	0,014035	0,0625	0,112281	0,126482	16,20429	0,014202	1,011858	
85,5	16	0,011696	0,0625	0,093567	0,103226	16,14131	0,009659	0,825806	
95	16	0,010526	0,0625	0,084211	0,091954	16,11427	0,007743	0,735632	

Di seguito, cerchiate in rosso, sono inserite le analisi delle prestazioni dei task 4.2 e 4.4 che permettono di mettere in evidenza la formazione dei colli di bottiglia.

TASK ASSEMBLAGGIO RACCHETTA MM1									
1/lambda	1/mu	lambda	mu	rho	L	W	Lq	Wq	
9,5	16	0,105263	0,0625	1,684211	-2,46154	-23,3846	-4,14575	-39,3846	
14,25	16	0,070175	0,0625	1,122807	-9,14286	-130,286	-10,2657	-146,286	
19	16	0,052632	0,0625	0,842105	5,333333	101,3333	4,491228	85,33333	
23,75	16	0,042105	0,0625	0,673684	2,064516	49,03226	1,390832	33,03226	
28,5	16	0,035088	0,0625	0,561404	1,28	36,48	0,718596	20,48	
42,75	16	0,023392	0,0625	0,374269	0,598131	25,57009	0,223862	9,570093	
57	16	0,017544	0,0625	0,280702	0,390244	22,2439	0,109542	6,243902	
71,25	16	0,014035	0,0625	0,224561	0,289593	20,63348	0,065031	4,633484	
85,5	16	0,011696	0,0625	0,187135	0,230216	19,68345	0,043081	3,683453	
95	16	0,010526	0,0625	0,168421	0,202532	19,24051	0,034111	3,240506	

TASK REALIZZAZIONE TELAIO MM1									
1/lambda	1/mu	lambda	mu	rho	L	W	Lq	Wq	
9,5	16	0,105263	0,0625	1,684211	-2,46154	-23,3846	-4,14575	-39,3846	
14,25	16	0,070175	0,0625	1,122807	-9,14286	-130,286	-10,2657	-146,286	
19	16	0,052632	0,0625	0,842105	5,333333	101,3333	4,491228	85,33333	
23,75	16	0,042105	0,0625	0,673684	2,064516	49,03226	1,390832	33,03226	
28,5	16	0,035088	0,0625	0,561404	1,28	36,48	0,718596	20,48	
42,75	16	0,023392	0,0625	0,374269	0,598131	25,57009	0,223862	9,570093	
57	16	0,017544	0,0625	0,280702	0,390244	22,2439	0,109542	6,243902	
71,25	16	0,014035	0,0625	0,224561	0,289593	20,63348	0,065031	4,633484	
85,5	16	0,011696	0,0625	0,187135	0,230216	19,68345	0,043081	3,683453	
95	16	0,010526	0,0625	0,168421	0,202532	19,24051	0,034111	3,240506	

Dalle precedenti tabelle emerge, che per quanto riguarda il Task 4.2 e 4.4 il *numero di clienti in coda (Lq)* è 101,3333 e il *tempo di attesa in coda (Wq)* è 85,3333, entrambi molto elevati, indicando una congestione significativa e lunghi tempi di attesa, inoltre, notiamo come anche i rho risultano elevati. Diversamente, a ciò che accade con code MM2 dove Lq e Wq sono rispettivamente pari a 19,44781 e 5,818182 e i rho sono pari esattamente alla metà (*evidenziati in verde*).

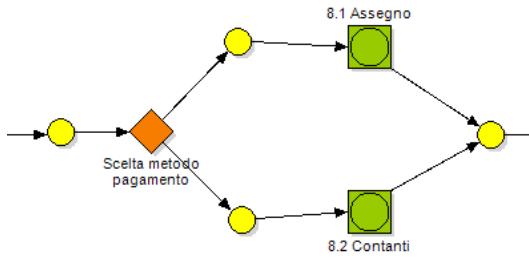
La configurazione con due serventi (MM2) risulta preferibile per i task critici come la *realizzazione del telaio* (4.2) e *l'assemblaggio della racchetta* (4.4), poiché riduce i tempi di attesa in coda (Wq), migliorando anche l'efficienza complessiva del sistema.

Di seguito abbiamo le tabelle che indicano il Work Time complessivo dei Task 4.2 e 4.3, uno considerando una coda di tipo MM2 nel Task 4.2 e l'altra considerando una coda di MM1 nel medesimo task.

MM1	
W And	
-212,417	-192,563
43,12387	-141,905
30,43624	104,9732
26,4946	53,53188
24,60985	41,05405
22,33052	29,98171
21,49143	26,52303
21,0654	24,82731
20,81055	23,81931
20,69141	23,34721

I valori contenuti nella precedente tabella rappresentano il Work Time complessivo dei due rami paralleli, calcolati come $W4.2 + W4.3 - 1 / (1/W4.2) + (1/W4.3)$. Il Work Time molto elevato del Task 4.2 in caso di coda MM1, fa sì che anche il Work Time complessivo dei due rami paralleli sia elevato.

Costrutto di selezione (XOR Split)



Una volta usciti dal ciclo abbiamo una serie di costrutti sequenziali per poi arrivare al costrutto di selezione, che è stato utilizzato per gestire il pagamento.

In particolare, tramite lo XOR, viene modellata la possibilità che l'azienda dà ai clienti di pagare tramite assegno (task 8.1 con una probabilità di 0.50) o contanti (task 8.2 anch'esso con una probabilità di 0.50).

TASK ASSEGNO									
1/lambda	1/mu	lambda	mu	rho	L	W	Lq	Wq	
33,68421	3	0,029688	0,333333	0,089063	0,09777	3,29331	0,008708	0,29331	
33,68421	3	0,029688	0,333333	0,089063	0,09777	3,29331	0,008708	0,29331	
40	3	0,025	0,333333	0,075	0,081081	3,243243	0,006081	0,243243	
50	3	0,02	0,333333	0,06	0,06383	3,191489	0,00383	0,191489	
60	3	0,016667	0,333333	0,05	0,052632	3,157895	0,002632	0,157895	
90	3	0,011111	0,333333	0,033333	0,034483	3,103448	0,001149	0,103448	
120	3	0,008333	0,333333	0,025	0,025641	3,076923	0,000641	0,076923	
150	3	0,006667	0,333333	0,02	0,020408	3,061224	0,000408	0,061224	
180	3	0,005556	0,333333	0,016667	0,016949	3,050847	0,000282	0,050847	
200	3	0,005	0,333333	0,015	0,015228	3,045685	0,000228	0,045685	

TASK CONTANTI									
1/lambda	1/mu	lambda	mu	rho	L	W	Lq	Wq	
33,68421	3	0,029688	0,333333	0,089063	0,09777	3,29331	0,008708	0,29331	
33,68421	3	0,029688	0,333333	0,089063	0,09777	3,29331	0,008708	0,29331	
40	3	0,025	0,333333	0,075	0,081081	3,243243	0,006081	0,243243	
50	3	0,02	0,333333	0,06	0,06383	3,191489	0,00383	0,191489	
60	3	0,016667	0,333333	0,05	0,052632	3,157895	0,002632	0,157895	
90	3	0,011111	0,333333	0,033333	0,034483	3,103448	0,001149	0,103448	
120	3	0,008333	0,333333	0,025	0,025641	3,076923	0,000641	0,076923	
150	3	0,006667	0,333333	0,02	0,020408	3,061224	0,000408	0,061224	
180	3	0,005556	0,333333	0,016667	0,016949	3,050847	0,000282	0,050847	
200	3	0,005	0,333333	0,015	0,015228	3,045685	0,000228	0,045685	

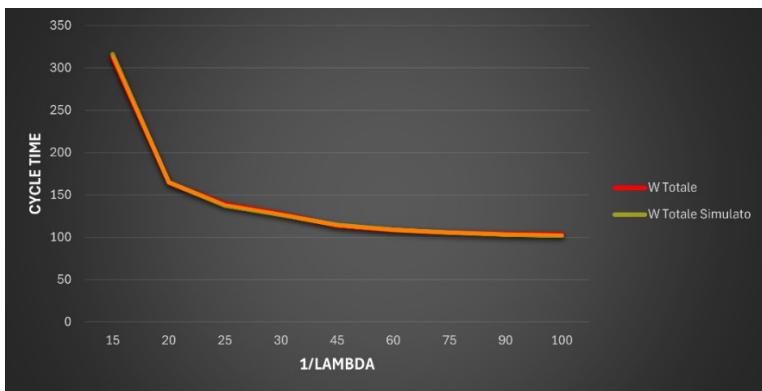
In questo caso il rapporto $1/\lambda$ di ogni task è raddoppiato (passa da 20 a 40) rispetto al rapporto iniziale in quanto, come specificato in precedenza, la probabilità che il cliente paghi con contanti o assegno è stata stimata in 0.5, perciò si è fatto un ulteriore rapporto tra il tempo di interarrivo e la probabilità di entrare in uno dei due rami alternativi.

Analisi Cycle Time

L'analisi del *Cycle Time* valuta il tempo complessivo necessario per completare tutti i task del processo, inclusi i task sequenziali e i rami di forward e di backward, ponderati in base alla probabilità di entrare in ciascun ramo.

4.2 e 4.4 MM2

1/LAMBDA	W Totale	W Totale Simulato
10	#DIV/0!	3199,09
15	312,242793	315,91
20	164,570777	164,99
25	138,9715753	137,23
30	127,485845	126,46
45	113,6386838	114,57
60	108,3690988	108,98
75	105,6011364	105,38
90	103,8986751	102,94
100	103,0871489	101,83



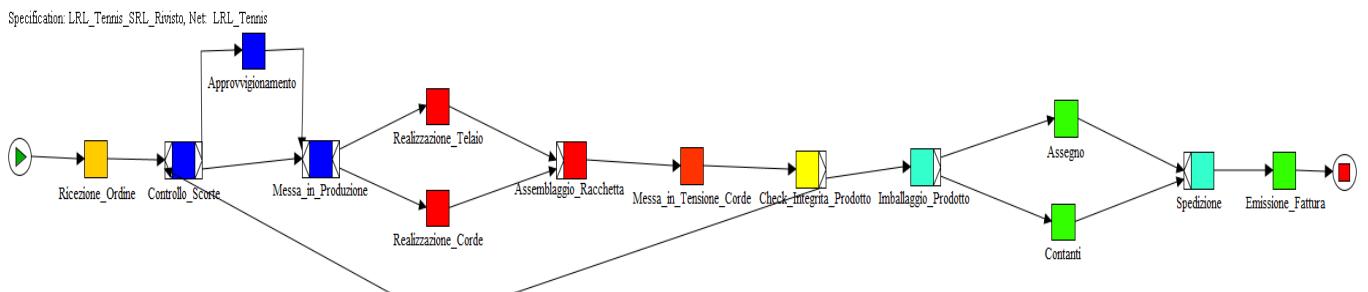
Sulla tabella e sul grafico sono riportati i valori trovati attraverso l'analisi delle prestazioni (*teorici*) e la media dei valori trovati simulando il processo su Jasper (*simulati*).

resource utilization	% busy
rolename	
Responsabile controllo qualità	66,95
Operaio 2	51,91
Commesso 1	49,6
Operaio 1	41,72
Operaio 3	41,3
Operaio 6	41,13
Operaio 5	41,06
Capo contabile	39,07
Capo magazziniere	29,74
Magazziniere 1	26,21
Operaio 4	26,19
Addetto spedizioni 2	24,72
Addetto spedizioni 1	14,9
Contabile	12,29

Notiamo come, aggiungendo i due operai nei task 4.2 e 4.4, i rho sono tutti accettabili (minori di 0.7), perciò l'azienda non avrebbe bisogno di assumere altri dipendenti.

Yawl

Lo stesso processo è stato modellato sul software YAWL che ci permette anche la gestione delle risorse.



Variabili di Rete

Data Variables for Net LRL_Tennis

	Name	Type	Scope	Initial Value
►	Nome	string	Local	
►	Cognome	string	Local	
►	Indirizzo_di_Spedizione	string	Local	
►	Data_Ordine	dateTime	Local	
►	Numero_Ordine	ID	Local	
►	Modello	string	Local	
►	Materiali_non_Disponibili	boolean	Local	<input type="checkbox"/>
►	Materiali_da_Ordinare	string	Local	
►	Tensione_Corde_kg	int	Local	0
►	Check_Superato	boolean	Local	<input type="checkbox"/>
►	Importo	float	Local	0
►	Assegno	boolean	Local	<input type="checkbox"/>
►	Data_di_Spedizione	dateTime	Local	
►	Numero_Fattura	ID	Local	
►	Data_Fattura	dateTime	Local	

+ - ^ v #

Cancel Apply OK

Decomposition Variables

	Name	Type	Scope	Default Output Value
►	Data_Ordine	dateTime	Input	###
►	Numero_Ordine	ID	Input	###
►	Modello	string	Input	###
►	Materiali_non_Disponibili	boolean	Output	<input type="checkbox"/>
►	Materiali_da_Ordinare	string	Output	

+ - ^ v # 🔍 ⌂

Nel task “Controllo scorte” è stata inserita la variabile di task (e di conseguenza anche di rete) Materiali_non_Disponibili di tipo Boolean, in modo tale da modellare lo XOR.

Target	Predicate
Approvvigionamento	LRL_Tennis/Materiali_non_Disponibili/text() = 'true'
Messa_in_Produzione	true()

Successivamente è stato modellato lo split predicates per stabilire secondo quale condizione il processo deve proseguire direttamente verso il task “Messa in produzione” o in alternativa passare per il task “Approvvigionamento”.

Decomposition Variables				
	Name	Type	Scope	Default Output Value
►	Data_Ordine	dateTime	Input	###
►	Numero_Ordine	ID	Input	###
►	Modello	string	Input	###
►	Tensione_Corde_kg	int	Input	###
►	Check_Superato	boolean	Output	□

+ - ^ v ↶ ↷ 🔍 ⌚ ⟳ ☰

Nel task “Check integrità prodotto” è stata inserita la variabile booleana “Check_Superato” per modellare lo XOR e l’eventuale ramo di backward del ciclo.

Target	Predicate
Imballaggio_Prodotto	LRL_Tennis/Check_Superato/text() = 'true'
Controllo_Scorte	true()

In questo caso lo split predicates è stato modellato in modo da far proseguire il processo verso il task “Imballaggio_Prodotto”. Mentre nel caso di “Check_Superato” == FALSE il processo entra nel ramo di backward e si attiva il ciclo a partire dal task “Controllo_Scorte”.

Decomposition Variables				
	Name	Type	Scope	Default Output Value
►	Nome	string	Output	
►	Cognome	string	Output	
►	Indirizzo_di_Spedizione	string	Output	
►	Data_Ordine	dateTime	Output	
►	Numero_Ordine	ID	Output	
►	Modello	string	Output	
►	Importo	float	Output	
►	Assegno	boolean	Output	□

+ - ^ v → ← 🔍 ⌂ ⌂

Nel primo Task “Recezione Ordine” sono state introdotte le varie variabili di Task per che interessa l’intero ordine, in particolare, è stata inserita una variabile (Assegno) ti tipo Booleano per poter indirizzare il flusso del processo verso uno dei due metodi di pagamento disponibile secondo specifiche condizioni.

Target	Predicate
Assegno	LRL_Tennis/Assegno/text() = 'true'
Contanti	true()

The bottom-most flow is the default

Nel caso di variabile booleana “Assegno” == TRUE il processo prosegue verso l’omonimo task.
In caso contrario si prosegue verso il task “Contanti”.

Gestione delle risorse

Di seguito sono riportati i ruoli, le risorse e i task di competenza di ciascuno.

Ruolo	Risorsa	Task
Commesso	Ilaria	1 Ricezione Ordine
Magazziniere	Barbara; Carlo	2 Controllo Scorte
Capo Magazziniere	Rita	3 Approvvigionamento; 4.1 Messa in produzione
Operaio	Simone Bianchi; Luca Neri; Matteo Verdi; Giulia; Maria; Umberto	4.2 Realizzazione telaio; 4.3 Realizzazione corde; 4.4 Assemblaggio Racchetta; 5 Messa in tensione corde
Responsabile controllo qualità	Giuseppe Rossi	6 Check integrità prodotto
Addetto spedizione	Carmela; Giacomo	7 Imballaggio prodotto; 9 Spedizione
Contabile	Emma; Nicola	8.1 Assegno; 8.2 Contanti (+ task metodo pagamento in YAWL)
Capo contabile	Stefano	10 Emissione Fattura

SIMULAZIONE PROCESSO

Edit Work Item

3.1: Ricezione Ordine

Nome	Cognome
Marco	Ranieri
Indirizzo_di_Spedizione	Data_Ordine
Via Pietro Bucci, 1	21/5/2024 12:06
Numero_Ordine	Modello
A1325	Rosso
Importo	<input checked="" type="checkbox"/> Assegno
215.00	

La variabile Booleana “Assegno”, successivamente, indirizzerà il flusso del processo nel Task Contanti (8.2)

Edit Work Item

2.2: Controllo Scorte

Data_Ordine	Numero_Ordine
15/5/2024 10:00	A1325
Modello	<input checked="" type="checkbox"/> Materiali_non_Disponibili
Rosso	
Materiali_da_Ordinare	

La variabile Booleana “Materiali non Disponibili”, se spuntata indirizzerà il flusso del processo nel Task del Processo in Approvvigionamento (3.)

Edit Work Item

2.4: Messa in Produzione

Data_Ordine
15/5/2024 10:00
Numero_Ordine
A1325
Modello
Rosso

Edit Work Item

2.3: Approvvigionamento

Materiali_da_Ordinare
Plastica

Edit Work Item

2.6:Realizzazione Telaio

Data_Ordine

15/5/2024



10:00



Numero_Ordine

A1325

Modello

Rosso

Edit Work Item

2.5:Realizzazione Corde

Data_Ordine

15/5/2024



10:00



Numero_Ordine

A1325

Modello

Rosso

Cancel

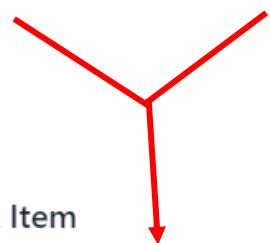
Save

Complete

Cancel

Save

Complete



Edit Work Item

2.7:Assemblaggio Racchetta

Data_Ordine

15/5/2024



10:00



Numero_Ordine

A1325

Modello

Rosso

Cancel

Save

Complete

Edit Work Item

2.8:Messa in Tensione Corde

Data_Ordine

15/5/2024



10:00



Numero_Ordine

A1325

Modello

Rosso

Tensione_Corde_kg

50

Cancel

Save

Complete

Edit Work Item

2.9:Check Integrita Prodotto

Data_Ordine

15/5/2024



10:00



Numero_Ordine

A1325

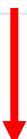
Modello

Rosso

Tensione_Corde_kg

50

Check_Superato



Cancel

Save

Complete

Questa Variabile Booleana se non spuntata fa ritornare il Flusso del processo in Controlle Scorte, attivando quindi il ramo di Backword del ciclo.

Edit Work Item

2.10:Imballaggio Prodotto

Data_Ordine

15/5/2024 10:00

Numero_Ordine

A1325

Modello

Rosso



Edit Work Item

2.11:Assegno

Nome

Marco

Cognome

Ranieri

Data_Ordine

15/5/2024 10:00

Numero_Ordine

A1325

Modello

Rosso

Importo

215.00

Edit Work Item

3.10:Contanti

Nome

Marco

Cognome

Ranieri

Data_Ordine

21/5/2024 12:06

Numero_Ordine

A1325

Modello

Rosso

Importo

215.00

Edit Work Item

2.12:Spedizione

Nome	Cognome
Marco	Ranieri
Data_Ordine	Numero_Ordine
15/5/2024 10:00	A1325
Modello	Importo
Rosso	215.00
Indirizzo_di_Spedizione	Data_di_Spedizione
Via Pietro Bucci, 1	15/5/2024 15:00

Cancel Save Complete

Edit Work Item

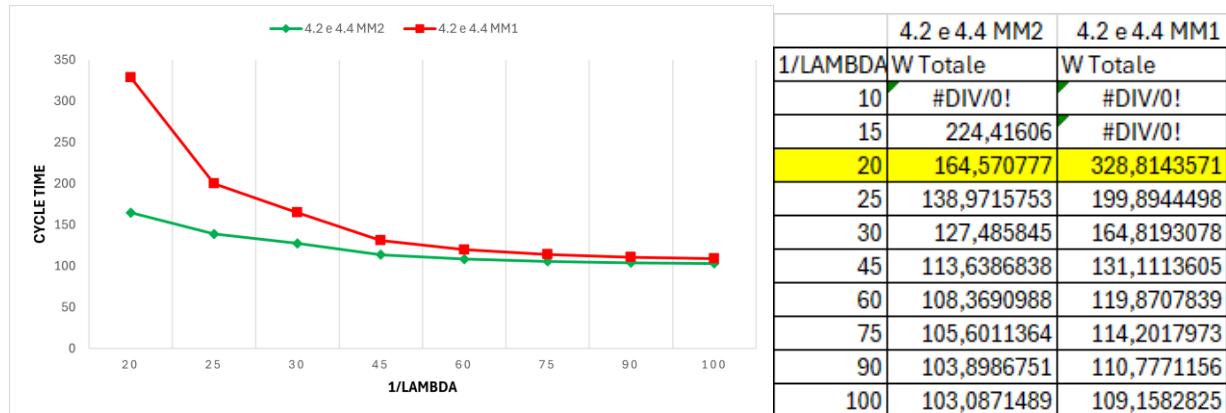
2.13:Emissione Fattura

Nome	Cognome
Marco	Ranieri
Indirizzo_di_Spedizione	Numero_Ordine
Via Pietro Bucci, 1	A1325
Modello	Importo
Rosso	215.00
<input checked="" type="checkbox"/> Assegno	Numero_Fattura
Data_Fattura	FA1325

Altre osservazioni

Come descritto in precedenza nella relazione, dalle prime simulazioni esplorative effettuate in Yasper e dai primi calcoli teorici dell'analisi delle prestazioni, si sono individuati dei colli bottiglia in posizione 4.2 e 4.4, per cui si è optato per l'inserimento di un ulteriore servente in entrambi i task (portandoli perciò da code M/M/1 a M/M/2).

Di seguito sono riportate la rappresentazione grafica e quella analitica della differenza tra il Cycle Time in caso di M/M/1 e M/M/2.



	4.2 e 4.4 MM2	4.2 e 4.4 MM1
1/LAMBDA	W Totale	W Totale
10	#DIV/0!	#DIV/0!
15	224,41606	#DIV/0!
20	164,570777	328,8143571
25	138,9715753	199,8944498
30	127,485845	164,8193078
45	113,6386838	131,1113605
60	108,3690988	119,8707839
75	105,6011364	114,2017973
90	103,8986751	110,7771156
100	103,0871489	109,1582825

Come si nota dal grafico, al diminuire del tempo di interarrivo la differenza tra il Cycle Time con coda MM2 e coda MM1 aumenta fino al divergere in maniera importante a tempi di interarrivo minori di 20.

Nella fase iniziale della modellazione del progetto, si sono riscontrati due colli di bottiglia.

show case list

report per emitore-collector pair

from	to	collected	completed	wait time	cycle time	work time	cost
(tr2)	(tr26)	9978	9978	222.33	323.23	102.77	115.18

time elapsed: 148401.73 # generated: 10000 # completed: 9978

resource utilization

rolename	% busy
Operaio 3	85.51
Operaio 1	83.46
Responsabile controllo qualità	68.17
Operaio 2	53.67
Commissario 1	49.70
Capo contabile	39.02
Capo magazziniere	30.19
Operaio 4	26.23
Magazziniere 1	26.04
Addetto spedizioni 2	25.16
Addetto spedizioni 1	15.19
Contabile	12.6

Come si deduce dall'immagine i tassi di utilizzo elevati di "Operaio 1" e "Operaio 3" testimoniano la posizione (task 4.2 e 4.4) in cui si formano i due colli di bottiglia, comportando elevati valori di Wait Time e di conseguenza dell'intero Cycle Time. Per mitigare questi problemi riscontrati, si è pensato di aggiungere un servente in più nei task critici.

Inizialmente sono state effettuate delle simulazioni inserendo un secondo servente nel task “Realizzazione Telaio”.

report per emitter-collector pair								time elapsed 31058,5	# generated 5000	# completed 4987
from	to	collected	completed	wait time	cycle time	work time	cost			
(tr2)	(tr26)	4987	4987	155,91	254,63	103,21	115,63			

resource utilization		% busy
rolename		
Operaio 3		84,66
Responsabile controllo qualità		68,48
Operaio 2		53,28
Commesso		50,31
Operaio 5		42,43
Operaio 1		41,98
Capo contabile		40,16
Magazziniere 1		26,27
Operaio 4		26,11
Magazziniere 2		26,02
Addetto spedizioni 2		25,08
Addetto spedizioni 1		15,04
Contabile 1		7,87
Contabile 2		7,58
Capo magazziniere		4,28

Successivamente le simulazioni sono state effettuate inserendo il secondo servente nel task “Assemblaggio Racchetta”.

report per emitter-collector pair								time elapsed 167761,24	# generated 10000	# completed 9978
from	to	collected	completed	wait time	cycle time	work time	cost			
(tr2)	(tr26)	9978	9978	143,22	244,4	102,76	115,17			

resource utilization		% busy
rolename		
Operaio 1		85,38
Responsabile controllo qualità		68,35
Operaio 2		52,52
Commesso 1		50,84
Operaio 5		41,91
Operaio 3		41,41
Capo contabile		39,5
Capo magazziniere		30,9
Magazziniere 1		26,62
Operaio 4		26,6
Addetto spedizioni 2		25,27
Addetto spedizioni 1		15,07
Contabile		12,41

In entrambi i casi il tasso di utilizzo del task critico diminuisce, ma i risultati su Wait Time e Cycle Time rimangono non soddisfacenti per cui si è deciso di utilizzare la coda di tipo MM2 per entrambi i task arrivando così a risultati migliori e apprezzabili.

<input type="checkbox"/> show case list		time elapsed 65603,48	# generated 10000	# completed 9997
report per emit-collector pair				
from (tr2)	to (tr26)	collected 9997	completed 9997	wait time 64,83
resource utilization				
rolename			% busy	
Responsabile controllo qualità			67,97	
Operario 2			52,49	
Commissario			49,68	
Operario 3			41,82	
Operario 5			41,65	
Operario 1			41,48	
Operario 6			41,33	
Capo contabile			39,94	
Magazziniere 2			26,09	
Magazziniere 1			25,81	
Operario 4			25,8	
Addetto spedizioni 2			25,01	
Addetto spedizioni 1			14,74	
Contabile 2			7,51	
Contabile 1			7,36	
Capo magazziniere			3,91	

Per migliorare ulteriormente questo processo, alcuni ruoli potrebbero essere uniti in modo da non avere serventi poco utilizzati. Da un'analisi sui rho, notiamo come Contabile 1 e Contabile 2 potrebbero essere uniti in una sola figura così come Capo Magazziniere e Magazziniere 2