

Progetto Processi di Business “Ingresso in Palestra”

Lavecchia Nicola

Introduzione

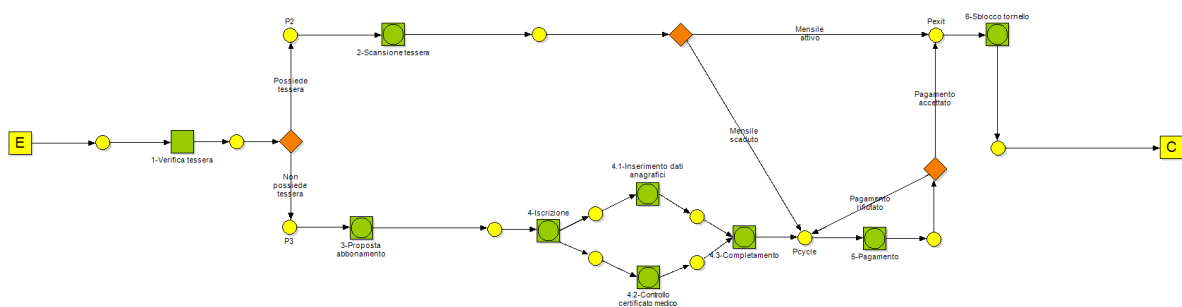
Una generica palestra desidera gestire le attività che intercorrono tra l'entrata alla reception e l'entrata in sala pesi.

In particolare si desidera che vengano gestite le iscrizioni per quelle persone che non sono in possesso di un abbonamento mentre si desidera che venga assicurata l'entrata a chi ne possiede uno nel minor tempo possibile. Si rende inoltre disponibile il rinnovo di un abbonamento per chi ne possiede uno scaduto tramite il solo pagamento.

Inizialmente si è creata una rappresentazione grafica del processo tramite il software Jasper, successivamente lo si è usato per un confronto tra i risultati derivanti dalla simulazione. Il processo viene poi ricreato nel software Yawl in modo da gestirlo da un diverso punto di vista e applicarlo in modo pratico.

Yasper

Ricorrendo al suddetto software si è sviluppata la seguente rete di Petri.



Si precisa che essendo le attività descritte da task che nella vita reale richiedono poco tempo ad essere svolte, si è deciso di impostare l'unità di misura temporale in secondi in modo da fornire più lavoro al sistema.

Al fine di trattare il processo in modo semplice dal punto di vista matematico si è adottata una distribuzione esponenziale per descrivere il processo. Ne consegue che media e deviazione

standard siano uguali nell'emittor e nei task. In particolare l'emittor è stato posto con valore $1/\lambda$ uguale a 600 secondi equivalenti a 10 minuti. Perciò entra in palestra una persona ogni 10 minuti.

Analizzandola nel dettaglio troviamo i seguenti task:

- **1-Verifica tessera:** Tempo medio di lavoro 30 secondi. Un/a Receptionist si occupa di verificare se il cliente che entra nella reception possiede la tessera o meno, nel primo caso il cliente procede da solo verso la macchina scanner, nel secondo caso lo si indirizza verso Dipendente 1.
 - **2-Scansione tessera:** Tempo medio di lavoro 30 secondi. Questo task ha una probabilità di scattare molto alta (80%) poiché la maggior parte delle persone che entrano in palestra in una data giornata possiede già un abbonamento per poter svolgere l'attività. Il task viene eseguito da una macchina scanner che si occupa in questo caso di verificare la validità della tessera. Nel 25% dei casi l'abbonamento risulterà scaduto e il cliente dovrà perciò provvedere al pagamento del rinnovo prima di entrare in sala.
 - **3-Proposta abbonamento:** Tempo medio di lavoro 360 secondi. Questo task ha una probabilità di scattare del 20%. Un servente che viene chiamato Dipendente 1 si occupa di esporre i 3 piani di abbonamenti che la palestra offre, in particolare il piano mensile, trimestrale e semestrale. Il cliente sceglie in questo passo per quanto tempo sottoscrivere un abbonamento.
 - **4-Iscrizione:** Tempo medio di lavoro 30 secondi. Dipendente 2 comunica al cliente l'importo mensile da pagare derivante dalla scelta del piano di abbonamento.
 - **4.1- Inserimento dati anagrafici:** Tempo medio di lavoro 240 secondi. Responsabile documenti 1 si occupa di rilevare i dati anagrafici del richiedente.
 - **4.2-Controllo certificato medico:** Tempo medio di lavoro 120 secondi. Responsabile documenti 2 si occupa di verificare il certificato presentato dal richiedente controllandone la veridicità.
- 4.1 e 4.2 sono task che vengono abilitati e scattano in parallelo.
- **4.3-Completamento:** Tempo medio di lavoro 120 secondi. Dipendente 3 si occupa di stampare il contratto contenente tutte le informazioni registrate fino a questo momento e di farlo firmare al cliente.
 - **5- Pagamento:** Tempo medio di lavoro 120 secondi. Il contabile si occupa del pagamento dell'importo concordato, il quale può essere convalidato o meno, con una probabilità del 25% si entrerà in un ciclo e il pagamento dovrà essere ritentato.

- **6-Sblocco tornello:** Tempo medio di lavoro 30 secondi. La macchina scanner si occupa di sbloccare il tornello dopo aver verificato che la tessera corrispondi ad un abbonamento attivo. Una volta sbloccato il tornello il processo si conclude.

Successivamente sono stati definiti i ruoli e ad ogni servente è stato assegnato un task

	role	capacity
►	Computer 1	1
	Computer 2	1
	Dipendente 1	1
	Responsabile documenti 1	1
	Contabile	1
	Responsabile documenti 2	1
	Dipendente 2	1
	Dipendente 3	1
	Receptionist	1

[illegible]

Analisi delle prestazioni

Sono state analizzate alcune componenti del workflow mediante il supporto del software Excel, e si sono quindi calcolate le seguenti misure

- ρ : utilizzazione dei server. Si considerano valori per cui ρ è minore di 1.
- L : numero medio di richieste nel sistema.
- Lq : numero medio di utenti in coda.
- T : tempo medio affinché entri una richiesta nel sistema.
- Tq : tempo medio di attesa nella coda.

I lambda (numero di casi medio previsto per unità di tempo), sono stati calcolati per ogni task con le seguenti formule

Emittor:	λ_E	TASK 4.1	$\lambda_{4.1} = \lambda_E * p_3$
TASK 1	$\lambda_1 = \lambda_E$	TASK 4.2	$\lambda_{4.2} = \lambda_E * p_3$
TASK 2	$\lambda_2 = \lambda_E * p_2$	TASK 4.3	$\lambda_{4.3} = \lambda_E * p_3$
TASK 3	$\lambda_3 = \lambda_E * p_3$	TASK 5	$\lambda_5 = [\lambda_E * p_3 * (1/(1-p_{cycle})) + \lambda_E * p_2 * (p_{cycle}/(1-p_{cycle}))]$
TASK 4	$\lambda_4 = \lambda_E * p_3$	TASK 6	$\lambda_6 = [\lambda_E * p_3 * (1/(1-p_{cycle})) + \lambda_E * p_2]$

pcycle	0,25	Emittor	
pexit	0,75	1/lambdaE	lambdaE
		1000	0,001
p2	0,8	800	0,00125
p3	0,2	600	0,001667
		400	0,0025
		300	0,003333
		200	0,005
		100	0,01
		90	0,011111
		80	0,0125
		70	0,014286
		60	0,016667

Verifica tessera

lambda1	1/mu1	mu1	rho1	L1	T1	Lq1	Tq1	Cost1
0,001	30	0,033333	0,03	0,03092784	30,9278351	0,000928	0,92783505	1
0,00125	30	0,033333	0,0375	0,03896104	31,1688312	0,001461	1,16883117	1
0,001667	30	0,033333	0,05	0,05263158	31,5789474	0,002632	1,57894737	1
0,0025	30	0,033333	0,075	0,08108108	32,4324324	0,006081	2,43243243	1
0,003333	30	0,033333	0,1	0,11111111	33,3333333	0,011111	3,33333333	1
0,005	30	0,033333	0,15	0,17647059	35,2941176	0,026471	5,29411765	1
0,01	30	0,033333	0,3	0,42857143	42,8571429	0,128571	12,8571429	1
0,011111	30	0,033333	0,333333	0,5	45	0,166667	15	1
0,0125	30	0,033333	0,375	0,6	48	0,225	18	1
0,014286	30	0,033333	0,428571	0,75	52,5	0,321429	22,5	1
0,016667	30	0,033333	0,5	1	60	0,5	30	1

Scansione tessera

lambda2	1/mu2	mu2	rho2	L2	T2	Lq2	Tq2	Cost2
0,0008	30	0,033333	0,024	0,025	30,738	0,00059	0,73770492	1
0,001	30	0,033333	0,03	0,031	30,928	0,000928	0,92783505	1
0,001333	30	0,033333	0,04	0,042	31,25	0,001667	1,25	1
0,002	30	0,033333	0,06	0,064	31,915	0,00383	1,91489362	1
0,002667	30	0,033333	0,08	0,087	32,609	0,006957	2,60869565	1
0,004	30	0,033333	0,12	0,136	34,091	0,016364	4,09090909	1
0,008	30	0,033333	0,24	0,316	39,474	0,075789	9,47368421	1
0,008889	30	0,033333	0,266667	0,364	40,909	0,09697	10,9090909	1
0,01	30	0,033333	0,3	0,429	42,857	0,128571	12,8571429	1
0,011429	30	0,033333	0,342857	0,522	45,652	0,178882	15,6521739	1
0,013333	30	0,033333	0,4	0,667	50	0,266667	20	1

Proposta abbonamento

lambda3	1/mu3	mu3	rho3	L3	T3	Lq3	Tq3	Cost3
0,0002	360	0,002778	0,072	0,07759	387,9	0,005586207	27,9310345	1
0,00025	360	0,002778	0,09	0,0989	395,6	0,008901099	35,6043956	1
0,000333333	360	0,002778	0,12	0,13636	409,1	0,016363636	49,0909091	1
0,0005	360	0,002778	0,18	0,21951	439	0,039512195	79,0243902	1
0,000666667	360	0,002778	0,24	0,31579	473,7	0,075789474	113,684211	1
0,001	360	0,002778	0,36	0,5625	562,5	0,2025	202,5	1
0,002	360	0,002778	0,72	2,57143	1286	1,851428571	925,714286	1
0,002222222	360	0,002778	0,8	4	1800	3,2	1440	1
0,0025	360	0,002778	0,9	9	3600	8,1	3240	1
0,002857143	360	0,002778	1,028571	-36	-12600	-37,0285714	-12960	1
0,003333333	360	0,002778	1,2	-6	-1800	-7,2	-2160	1

Iscrizione

lambda4	1/mu4	mu4	rho4	L4	T4	Lq4	Tq4	Cost4
0,0002	30	0,033333	0,006	0,006036	30,18109	3,62173E-05	0,181087	1
0,00025	30	0,033333	0,0075	0,007557	30,2267	5,66751E-05	0,2267	1
0,000333333	30	0,033333	0,01	0,010101	30,30303	0,00010101	0,30303	1
0,0005	30	0,033333	0,015	0,015228	30,45685	0,000228426	0,456853	1
0,000666667	30	0,033333	0,02	0,020408	30,61224	0,000408163	0,612245	1
0,001	30	0,033333	0,03	0,030928	30,92784	0,000927835	0,927835	1
0,002	30	0,033333	0,06	0,06383	31,91489	0,003829787	1,914894	1
0,002222222	30	0,033333	0,066667	0,071429	32,14286	0,004761905	2,142857	1
0,0025	30	0,033333	0,075	0,081081	32,43243	0,006081081	2,432432	1
0,00285714	30	0,033333	0,085714	0,09375	32,8125	0,008035714	2,8125	1
0,003333333	30	0,033333	0,1	0,111111	33,33333	0,011111111	3,333333	1

Inserimento dati anagrafici

lambda4.1	1/mu4.1	mu4.1	rho4.1	L4.1	T4.1	Lq4.1	Tq4.1	Cost4.1
0,0002	240	0,004167	0,048	0,05042	252,1008	0,00242	12,10084034	1
0,00025	240	0,004167	0,06	0,06383	255,3191	0,00383	15,31914894	1
0,000333	240	0,004167	0,08	0,086957	260,8696	0,006957	20,86956522	1
0,0005	240	0,004167	0,12	0,136364	272,7273	0,016364	32,72727273	1
0,000667	240	0,004167	0,16	0,190476	285,7143	0,030476	45,71428571	1
0,001	240	0,004167	0,24	0,315789	315,7895	0,075789	75,78947368	1
0,002	240	0,004167	0,48	0,923077	461,5385	0,443077	221,5384615	1
0,002222	240	0,004167	0,533333	1,142857	514,2857	0,609524	274,2857143	1
0,0025	240	0,004167	0,6	1,5	600	0,9	360	1
0,002857	240	0,004167	0,685714	2,181818	763,6364	1,496104	523,6363636	1
0,003333	240	0,004167	0,8	4	1200	3,2	960	1

Controllo certificato medico

lambda4.2	1/mu4.2	mu4.2	rho4.2	L4.2	T4.2	Lq4.2	Tq4.2	Cost4.2
0,0002	120	0,008333	0,024	0,02459	122,9508	0,00059	2,95082	1
0,00025	120	0,008333	0,03	0,030928	123,7113	0,000928	3,71134	1
0,000333333	120	0,008333	0,04	0,041667	125	0,001667	5	1
0,0005	120	0,008333	0,06	0,06383	127,6596	0,00383	7,659574	1
0,000666667	120	0,008333	0,08	0,086957	130,4348	0,006957	10,43478	1
0,001	120	0,008333	0,12	0,136364	136,3636	0,016364	16,36364	1
0,002	120	0,008333	0,24	0,315789	157,8947	0,075789	37,89474	1
0,002222222	120	0,008333	0,266667	0,363636	163,6364	0,09697	43,63636	1
0,0025	120	0,008333	0,3	0,428571	171,4286	0,128571	51,42857	1
0,002857143	120	0,008333	0,342857	0,521739	182,6087	0,178882	62,6087	1
0,003333333	120	0,008333	0,4	0,666667	200	0,266667	80	1

Completamento

lambda4.3	1/mu4.3	mu4.3	rho4.3	L4.3	T4.3	Lq4.3	Tq4.3	Cost4.3
0,0002	120	0,008333	0,024	0,02459	122,9508	0,00059	2,95082	1
0,00025	120	0,008333	0,03	0,030928	123,7113	0,000928	3,71134	1
0,000333	120	0,008333	0,04	0,041667	125	0,001667	5	1
0,0005	120	0,008333	0,06	0,06383	127,6596	0,00383	7,659574	1
0,000667	120	0,008333	0,08	0,086957	130,4348	0,006957	10,43478	1
0,001	120	0,008333	0,12	0,136364	136,3636	0,016364	16,36364	1
0,002	120	0,008333	0,24	0,315789	157,8947	0,075789	37,89474	1
0,002222	120	0,008333	0,266667	0,363636	163,6364	0,09697	43,63636	1
0,0025	120	0,008333	0,3	0,428571	171,4286	0,128571	51,42857	1
0,002857	120	0,008333	0,342857	0,521739	182,6087	0,178882	62,6087	1
0,003333	120	0,008333	0,4	0,666667	200	0,266667	80	1

Pagamento

lambda5	1/mu5	mu5	rho5	L5	T5	Lq5	Tq5	Cost5
0,000533	120	0,008333	0,064	0,068376	128,2051	0,004376	8,205128	1
0,000667	120	0,008333	0,08	0,086957	130,4348	0,006957	10,43478	1
0,000889	120	0,008333	0,106667	0,119403	134,3284	0,012736	14,32836	1
0,001333	120	0,008333	0,16	0,190476	142,8571	0,030476	22,85714	1
0,001778	120	0,008333	0,213333	0,271186	152,5424	0,057853	32,54237	1
0,002667	120	0,008333	0,32	0,470588	176,4706	0,150588	56,47059	1
0,005333	120	0,008333	0,64	1,777778	333,3333	1,137778	213,3333	1
0,005926	120	0,008333	0,711111	2,461538	415,3846	1,750427	295,3846	1
0,006667	120	0,008333	0,8	4	600	3,2	480	1
0,007619	120	0,008333	0,914286	10,66667	1400	9,752381	1280	1
0,008889	120	0,008333	1,066667	-16	-1800	-17,0667	-1920	1

Sblocco tornello

lambda6	1/mu6	mu6	rho6	L6	T6	Lq6	Tq6	Cost6
0,001067	30	0,033333	0,032	0,033058	30,99174	0,001058	0,99173554	1
0,001333	30	0,033333	0,04	0,041667	31,25	0,001667	1,25	1
0,001778	30	0,033333	0,053333	0,056338	31,69014	0,003005	1,69014085	1
0,002667	30	0,033333	0,08	0,086957	32,6087	0,006957	2,60869565	1
0,003556	30	0,033333	0,106667	0,119403	33,58209	0,012736	3,58208955	1
0,005333	30	0,033333	0,16	0,190476	35,71429	0,030476	5,71428571	1
0,010667	30	0,033333	0,32	0,470588	44,11765	0,150588	14,1176471	1
0,011852	30	0,033333	0,355556	0,551724	46,55172	0,196169	16,5517241	1
0,013333	30	0,033333	0,4	0,666667	50	0,266667	20	1
0,015238	30	0,033333	0,457143	0,842105	55,26316	0,384962	25,2631579	1
0,017778	30	0,033333	0,533333	1,142857	64,28571	0,609524	34,2857143	1

Tempo totale

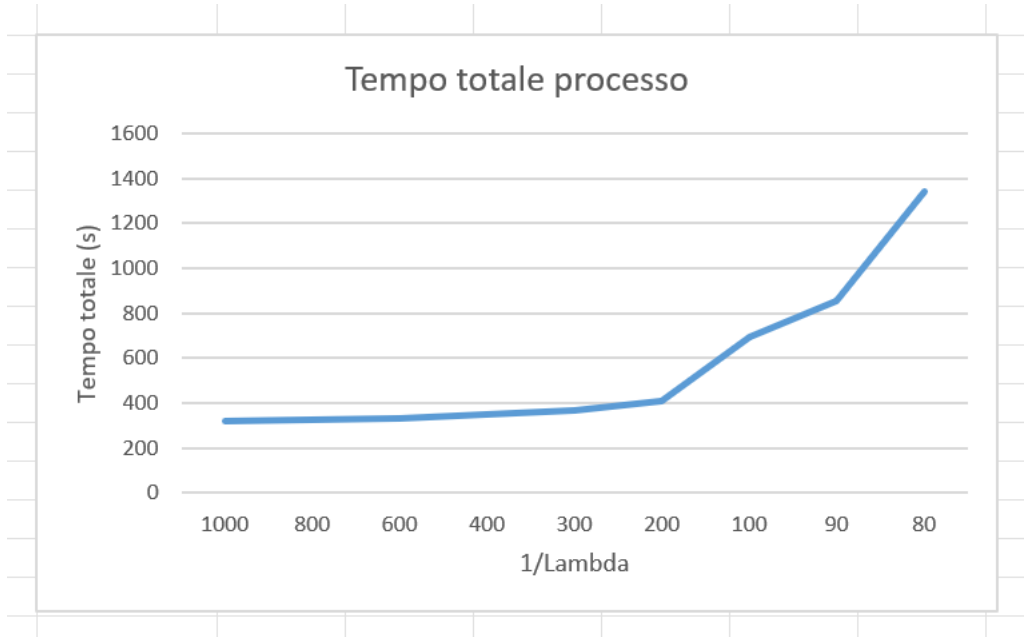
Nback	Nforw	Tparallelo	Tramo1	Tramo2	Ttot
0,333333	1,333333	292,407	200,882	58,7782	321,5798
0,333333	1,333333	295,6972	203,8305	59,52488	325,7742
0,333333	1,333333	301,3625	208,9722	60,8209	333,0622
0,333333	1,333333	313,4303	220,2095	63,62715	348,8777
0,333333	1,333333	326,5968	232,9436	66,76492	366,6239
0,333333	1,333333	356,915	264,4001	74,33155	409,7401
0,333333	1,333333	501,7861	484,3509	120,4678	691,7935
0,333333	1,333333	553,7841	620,6819	143,4965	855,7301
0,333333	1,333333	638,0952	1048,391	194,2857	1340,677
0,333333	1,333333	798,8766	-1943,81	409,8551	-1426,19
0,333333	1,333333	1228,571	-547,619	-440	-863,333

Costo totale

[illegible]

Si può notare che in alcune celle, le quali corrispondono a valori bassi di $1/\lambda$, il valore del ρ calcolato risulta essere superiore ad 1. Di conseguenza i valori calcolati a partire da questi valori potrebbero risultare negativi e non ha senso prenderli in considerazione.

Si procede ora con una rappresentazione grafica e in formula del tempo totale di esecuzione del processo:



Dal grafico che mette in relazione il valore $1/\lambda$ con il tempo totale di esecuzione si può notare come il tempo di esecuzione del processo aumenti rapidamente per valori di λ inferiori a 200, cioè quando nel sistema entra un caso con frequenza minore ai 200 secondi.

Per calcolare il tempo totale il processo è stato diviso in due rami:

$$N_{Forw} = \frac{1}{(1 - P_{CYCLE})}$$

$$T_{RAMO 1} = P_3 * \left\{ T3 + T4 + \left[T4.1 + T4.2 - \left(\frac{1}{\left(\frac{1}{T4.1} \right) + \left(\frac{1}{T4.2} \right)} \right) \right] + T4.3 + (T5 * N_{FORW}) \right\}$$

$$T_{RAMO 2} = P_2 * [T2 + (T5 * N_{FORW} * P_{CYCLE})]$$

$$T_{TOTALE} = T1 + T_{RAMO\ 1} + T_{RAMO\ 2} + T6$$

Per confermare la correttezza dei risultati derivanti dall'analisi matematica, si confrontano i risultati ottenuti con la simulazione in Jasper. Ci si aspetta che i valori siano vicini a quelli calcolati. In particolare i seguenti valori sono stati ottenuti simulando il processo con 3000 unità entranti nel sistema con un valore di 1/lambda pari a 600.

Serventi	Rho ottenuti tramite foglio di calcolo(%)	Rho ottenuti tramite simulazione Yasper(%)
Dipendente 1:	12	12
Contabile:	10,6	10,6
Responsabile documenti 1:	8	7,74
Receptionist:	5	5,26
Computer 2:	5,33	5,21
Responsabile documenti 2:	4	4,21
Computer 1:	4	4,16
Dipendente 3:	4	3,9
Dipendente 2:	1	0,96

rolename	% busy
Dipendente 1	12
Contabile	10,6
Responsabile documenti 1	7,74
Receptionist	5,26
Computer 2	5,21
Responsabile documenti 2	4,21
Computer 1	4,16
Dipendente 3	3,9
Dipendente 2	0,96

time elapsed			# generated	# completed
1434732			3000	2999
cycle time	work time	cost		
327,59	319,14	4,11		

Tempo e costo totale di esecuzione in corrispondenza del valore $1/\lambda$ pari a 600:

T totale	C totale
333,0622	4,233333

Si nota come sia i valori ρ che tempo e costo totale coincidono in media.

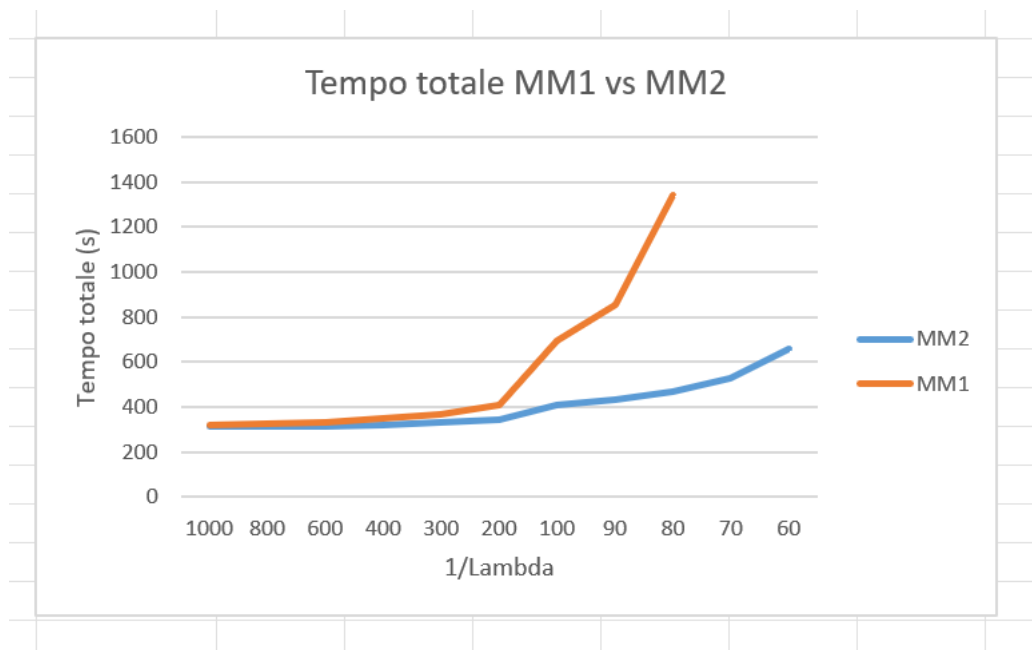
MM2 ed efficienza

Si è notato che in corrispondenza dei valori $1/\lambda$ pari a 60 e 70, i ρ dei task 3 e 5 sono maggiori di 1, in quei casi il processo non è stabile.

In modo da migliorare la situazione si è deciso di aggiungere un servente ai task 3 e 5 passando da MM1 a MM2.

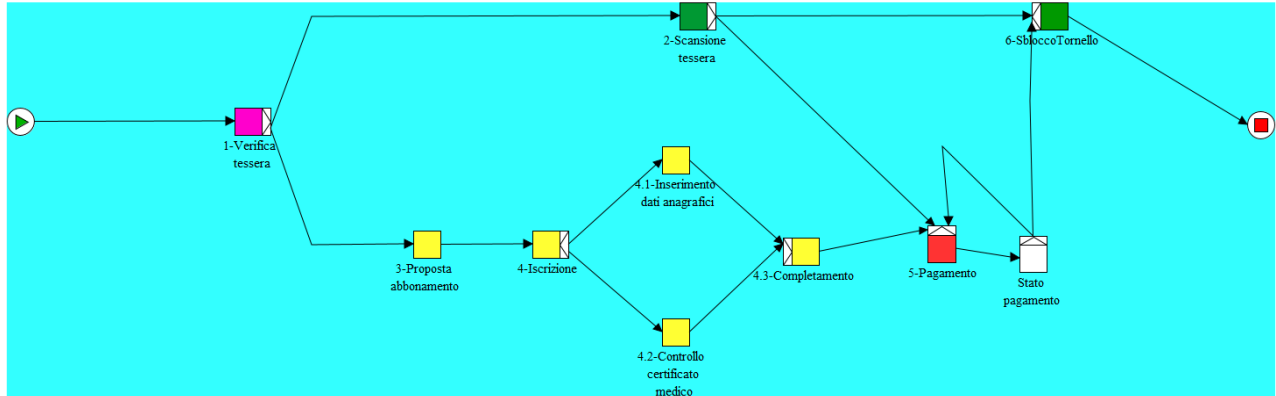
La soluzione adottata porta a risultati stabili ed efficienti, infatti i ρ sono analiticamente tutti inferiori ad 1 e inoltre il tempo totale dell'intero processo diminuisce.

Si può perciò concludere che nella coda MM1 in situazioni di sovraccarico i tempi crescono molto rapidamente per valori di $1/\lambda$ minori di 200, mentre nella coda MM2 i tempi crescono sempre a partire da quel valore ma in modo molto più contenuto come è possibile vedere dal grafico dei tempi a confronto.

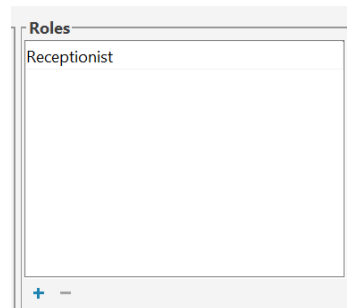
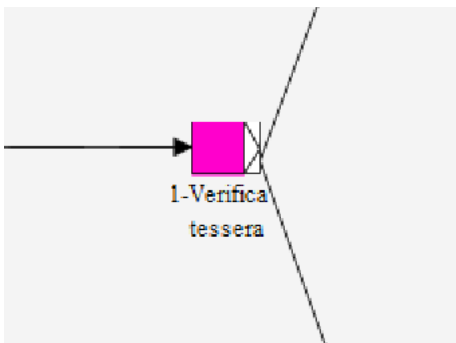


Yawl

Specification: Progetto_Nick, Net, Net



1- Verifica Tessera

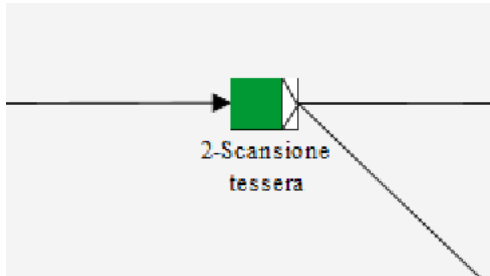


Decomposition Variables				
	Name	Type	Scope	Default Output Value
►	HaTessera	boolean	Output	<input type="checkbox"/>

Target	Predicate
_3-Proposta_abbonamento	/Net/HaTessera/text()='false'
_2-Scansione_tessera	true()

Nel primo task del processo, un/a receptionist verifica che il cliente in entrata sia in possesso di una tessera. Questo task presenta uno XOR Split Type che si traduce in due flussi in uscita associati ad una condizione booleana. In particolare, come è possibile vedere dagli Split Predicates, se il cliente non possiede una tessera allora questa variabile avrà valore 'false' e il cliente sarà indirizzato verso il ramo preposto alla sottoscrizione di un abbonamento, altrimenti si otterrà la variabile assumerà il valore standard che in Yawl equivale a quello più in basso. Lo svolgimento di questa mansione non richiede nessuna capacità in quanto si è pensato che qualunque dipendente, nella vita reale, sia in grado di svolgerla.

2- Scansione Tessera



Roles

Scanner tessera

Decomposition Variables				
	Name	Type	Scope	Default Output Value
▶	TesseraValida	boolean	Output	<input type="checkbox"/>

Filters

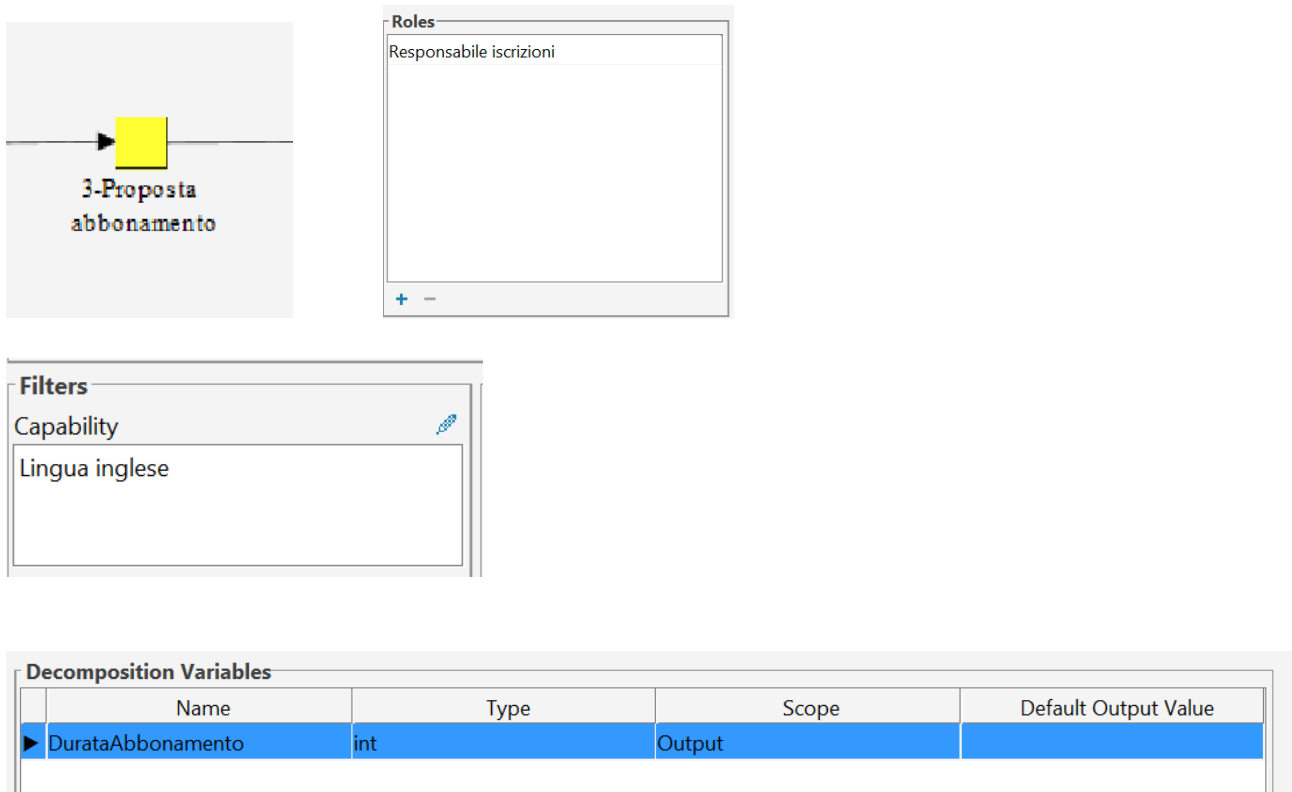
Capability

Capacità di calcolo

Target	Predicate
6-SbloccoTornello	/Net/TesseraValida/text()='true'
_5-Pagamento	true()

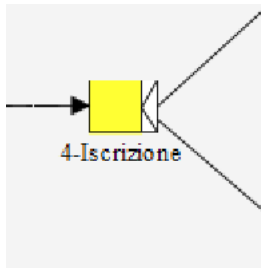
Se il cliente possiede la tessera, viene indirizzato verso questo task. Questo viene svolto dallo scanner tessera, che per poter analizzare la tessera ha bisogno di possedere una capacità di calcolo. In particolare lo scanner verifica che la tessera passata sia valida. A sua volta questo task presenta uno XOR Split Type con conseguente condizione booleana, se la condizione presenta il valore 'true' (tessera valida) allora si procede ad eseguire il task 6, dove la stessa macchina scanner procederà a sbloccare il tornello. Altrimenti, se la condizione presenta il valore 'false', significa che l'abbonamento del cliente è scaduto, verrà perciò indirizzato al task 5 relativo al pagamento per poter effettuare il rinnovo dello stesso.

3- Proposta Abbonamento



Il task 3 viene svolto se il cliente non possiede una tessera. Questo viene eseguito dal responsabile iscrizione al quale viene richiesta la conoscenza della lingua inglese in quanto svolge un'attività di comunicazione con il cliente che potrebbe non conoscere la lingua italiana. Infatti il compito l'addetta/o a questo task è di esporre i piani di abbonamento con relativi piani tariffari che variano a seconda della durata del contratto. Il cliente sceglie il tipo di abbonamento che desidera sottoscrivere. Viene prodotta una variabile DurataAbbonamento di tipo int.

4- Iscrizione



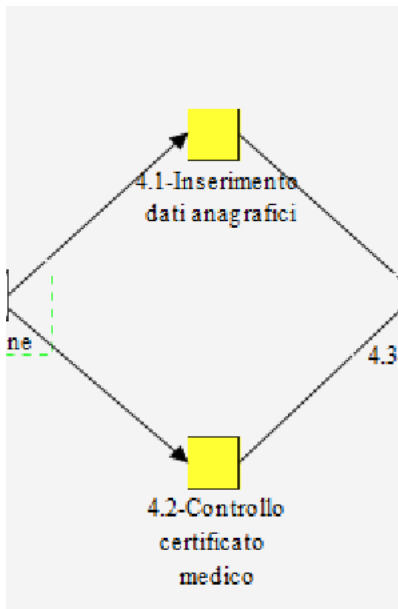
Filters
Capability
Lingua inglese

Roles
Responsabile iscrizioni

Decomposition Variables				
	Name	Type	Scope	Default Output Value
►	ImportoMensile	int	Output	

Questo Task dà inizio all'iscrizione effettiva del cliente. Esso viene svolto dal responsabile iscrizioni, al quale, come nel precedente task è richiesta la conoscenza della lingua inglese. Inoltre viene aggiunto un vincolo (principio di familiarità) sullo svolgimento del task, ovvero che sia lo/a stesso/a responsabile iscrizione che ha esposto i piani di abbonamento a iniziare l'iscrizione del nuovo cliente. Il Task 4 è caratterizzato da un AND Split Type e produce inoltre una variabile di tipo int che andrà riempita dal servente con l'ImportoMensile derivante dalla scelta del piano di abbonamento . Nello specifico i Task che seguono andranno eseguiti in parallelo, cioè contemporaneamente.

4.1- Inserimento dati anagrafici & 4.2- Controllo certificato medico



Roles	Net Parameters
Responsabile iscrizioni	

Constraints
<input checked="" type="checkbox"/> Choose completer(s) of task: _4-Iscrizione

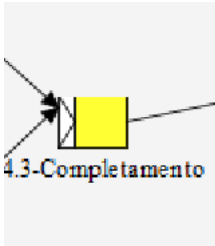
Decomposition variables				
	Name	Type	Scope	Default Output Value
►	DatiAnagrafici	string	Output	

Decomposition variables				
	Name	Type	Scope	Default Output Value
►	CertificatoMedico	YDocumentType	Output	

I Task 4.1 e 4.2 come detto in precedenza vengono svolti in parallelo dallo stesso Responsabile del task 4, infatti si è deciso di usare anche in questo caso il principio di familiarità. Il responsabile si occupa di rilevare i dati anagrafici del cliente e controlla che sia in possesso di un regolare certificato medico.

Entrambi i task producono una variabile in output, in particolare il Task 4.1 produce una variabile di tipo String DatiAnagrafici. Il Task 4.2 produce un documentType contenente il certificato medico che andrà poi inserito nel contratto del cliente.

4.3- Completamento



Roles

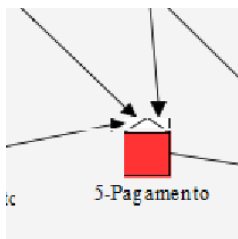
Responsabile iscrizioni

+ -

	Name	Type	Scope	Default Output Value
►	CertificatoMedico	YDocumentType	Input	###
►	DatiAnagrafici	string	Input	###
►	Contratto	YDocumentType	Output	
►	DurataAbbonamento	int	Input	###

Il task 4.3 è l'ultimo task svolto dal responsabile iscrizioni, anche in questo viene usato il principio di familiarità, per cui è lo stesso responsabile a svolgere tutte le mansioni che riguardano l'iscrizioni. Questo task è caratterizzato da un And Join Type e consiste nella stesura del contratto che verrà composto a partire dagli elementi accumulati nei task precedenti. Infatti, il Task 4.3 prende in input variabili già presenti in rete e le combina fornendo in output un documentType Contratto.

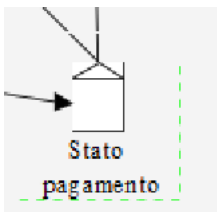
5- Pagamento



Roles	Capability
Contabile	Capacità economiche

Decomposition Variables			
	Name	Type	Scope
▶	DurataAbbonamento	int	Input
▶	ImportoMensile	int	Input
▶	ImportoTotale	int	Output

Il task 5 presenta uno XOR join Type, infatti può arrivare a questo task sia un cliente che si è appena iscritto ma anche un cliente che era già in possesso di un abbonamento, al fine di rinnovarlo. Questo task viene svolto dal contabile al quale vengono richieste capacità economiche. Inoltre il task prende le variabili DurataAbbonamento e ImportoMensile dalla rete e le moltiplica tra loro al fine di ottenere ImportoTotale. Quest'ultima è proprio la cifra che il cliente dovrà pagare per completare la fase di iscrizione o di rinnovo. Nel caso del rinnovo, il cliente pagherà la stessa cifra prevista dal contratto sottoscritto in precedenza.

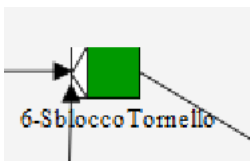


Per questioni di chiarezza, a differenza del processo modellato in Jasper si è deciso di inserire questo task. Stato pagamento in particolare, consiste nel verificare che il pagamento sia andato a buon fine. Infatti presenta uno XOR split type e una conseguente variabile booleana ValidazionePagamento. Quando questa variabile vale 'true' allora il pagamento si considera andato a buon fine e il cliente viene indirizzato direttamente allo sblocco del tornello senza passare per la verifica tessera in quanto sarà lo stesso contabile a provvedere alla sua apertura. Nel caso la variabile valga 'false', il pagamento è da considerarsi nullo, si tornerà perciò al Task 5 Pagamento.

Decomposition variables				
	Name	Type	Scope	Default Output Value
►	ValidazionePagamento	boolean	Output	<input type="checkbox"/>

Target	Predicate
6-SbloccoTornello	/Net/ValidazionePagamento/text()='true'
_5-Pagamento	true()

6- Sblocco Tornello



Roles	Net Parameters
Scanner tessera	
+ -	+ - ✎

Constraints

☒ Choose completer(s) of task: _2-Scansione_tessera ▼

Il task 6 presenta uno XOR join Type in quanto, come detto in precedenza, è possibile arrivare a questo task sia per un cliente che passa dal pagamento e sia per un cliente già in possesso di un regolare abbonamento. Sblocco Tornello viene svolto dallo stesso scanner tessera che svolge il Task 2, vige anche in questo caso il principio di familiarità. Esso non produce nessuna variabile. Questo Task termina il processo.