

Simulazione di scelte complesse nelle organizzazioni

PIETRO TERNA,

Dipartimento di Scienze economiche e finanziarie G. Prato dell'Università di Torino

pietro.terna@unito.it

<http://web.econ.unito.it/terna>

I wish your enterprise to-day may thrive.
Shakespeare, Julius Caesar

INTRODUZIONE

Si presenta la struttura di un modello di simulazione ad agenti indirizzato all'analisi di scelte complesse all'interno delle organizzazioni.

Alla base del modello sta il formalismo *jES*, *java Enterprise Simulator*, descritto in Terna (2002) e disponibile all'indirizzo web.econ.unito.it/terna/jes/ come piattaforma di simulazione fondata sull'uso della biblioteca di funzioni Swarm¹.

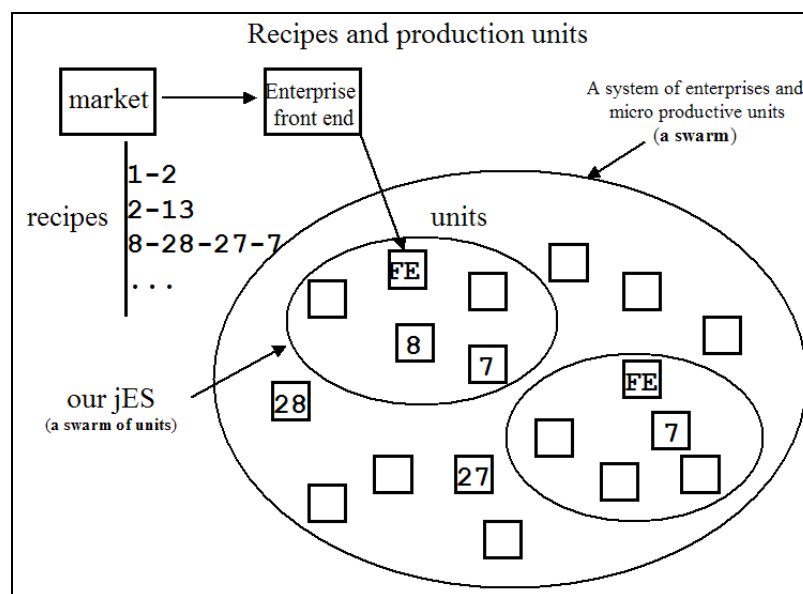


Figura 1. Una presentazione semplificata dei componenti di *jES*; le ricette sono presentate in forma semplificata, senza specificazione temporale dei singoli passi.

Il formalismo *jES* (vedere figura 1) consente tre differenti descrizioni del mondo: (i) il repertorio delle attività da svolgere, descritte secondo “ricette” che indicano i passi da compiere, la relativa sequenza, i tempi necessari, le eventuali caratteristiche speciali (nel caso dell'impresa le ricette descrivono gli ordini provenienti dal mercato o dall'interno dell'impresa stessa); (ii) la descrizione della struttura produttiva, tramite l'elencazione delle unità che possono svolgere i passi indicati nelle ricette sub (i), in modo esclusivo o

¹ wiki.swarm.org e www.swarm.org

no; questa descrizione può essere spinta sino al dettaglio desiderato e può riguardare una singola organizzazione o impresa, oppure un sistema (es. distretto) o filiera di produzione; (iii) la sequenza degli eventi nel tempo, con il lancio in successione di ricette tratte dal repertorio sub (i).

E' necessaria una precisazione sulla metodologia e sui suoi scopi: quello che si sta trattando non è un sistema di per sé orientato all'ottimizzazione, bensì alla riproduzione dell'esistente. Riprendendo le parole di Holland (2003), esattamente come esistono i simulatori di volo, è necessario costruire i simulatori di impresa.

Il simulatore di volo esegue ciò che l'utilizzatore decide; altrettanto il simulatore di impresa deve riprodurre ciò che il gestore di un "esperimento simulato" decide debba accadere, con le conseguenze positive o negative implicite nelle decisioni e rese esplicite dalla simulazione. Se nelle decisioni è compresa una strategia di ottimizzazione, il simulatore genererà le conseguenze delle scelte compiute.

UN FORMALISMO PER SIMULARE PROCESSI DI SCELTA: DALLE "COSE DA FARE" ...

In questa presentazione si sottolinea l'uso del formalismo sopra delineato, per simulare i processi di scelta, tramite i tre diversi sistemi definitivi.

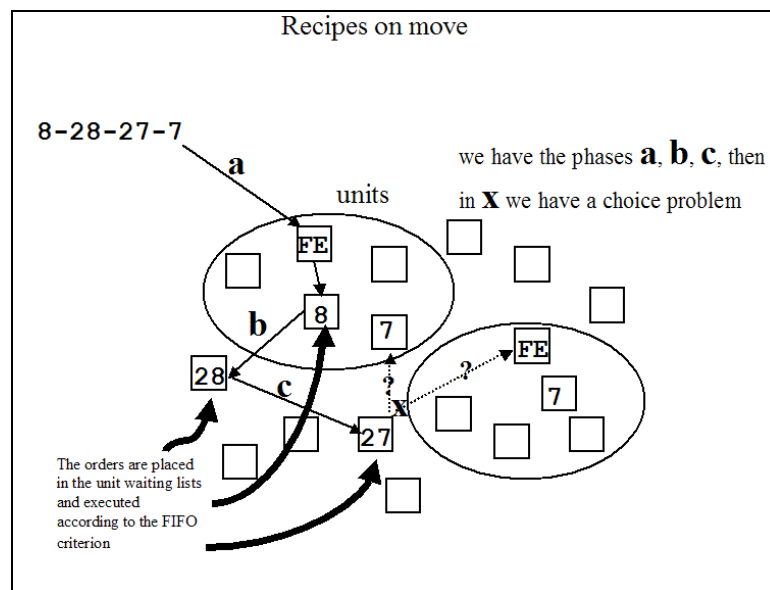


Figura 2. Una visione dinamica dei componenti di *jES*, con una scelta da compiere.

Il primo elemento del formalismo riguarda, infatti, la descrizione delle attività da svolgere, che sono una caratteristica comune di qualsiasi organizzazione, in ogni situazione in cui si debbano assumere delle decisioni.

Le ricette che definiscono le attività sono strutturate con una notazione molto ampia, che riporta l'indicazione delle cose da fare, degli ingredienti da utilizzare, dei tempi necessari, delle sequenze da rispettare, con tutti gli aspetti collaterali, dagli approvvigionamenti necessari alle attività da compiere in parallelo o in alternativa.

Nella figura 2 la sequenza della ricetta, sempre semplificata (passi da compiere 8, 28, 27, 7 con le unità capaci di svolgerli, propone un problema di scelta-decisione: dopo lo svolgimento del passo 27 a quale unità, tra le due disponibili, affidare il passo 7?

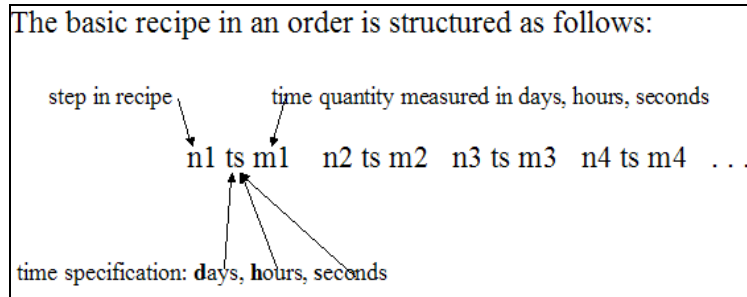


Figura 3. Una ricetta base.

Sin qui si è utilizzata una presentazione semplificata delle ricette; con maggiore realismo, nella figura 3 si presenta una sequenza di azioni da compiere, accompagnate dalle relative indicazioni temporali.

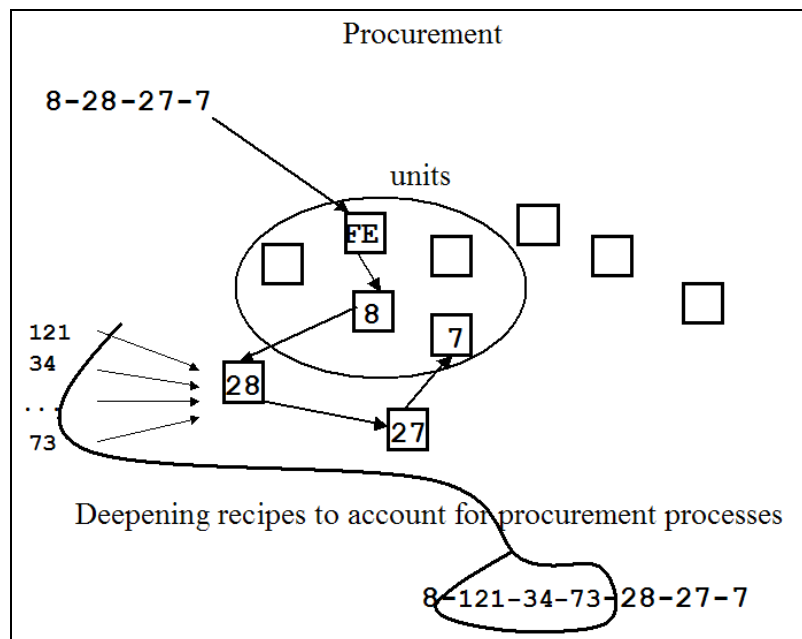


Figura 4. Il processo di approvvigionamento.

Alle fasi produttive può essere collegata l'indicazione degli approvvigionamenti necessari, intesi come produzioni interne od esterne rispetto al contesto organizzativo che si sta analizzando. Gli approvvigionamenti devono essere disponibili al momento in cui le fasi che li richiedono sono eseguite: per questo si predispongono, nell'ambito della simulazione, "magazzini reali o virtuali" in cui sono reperibili i beni approvvigionati, se ed in quanto siano tali.

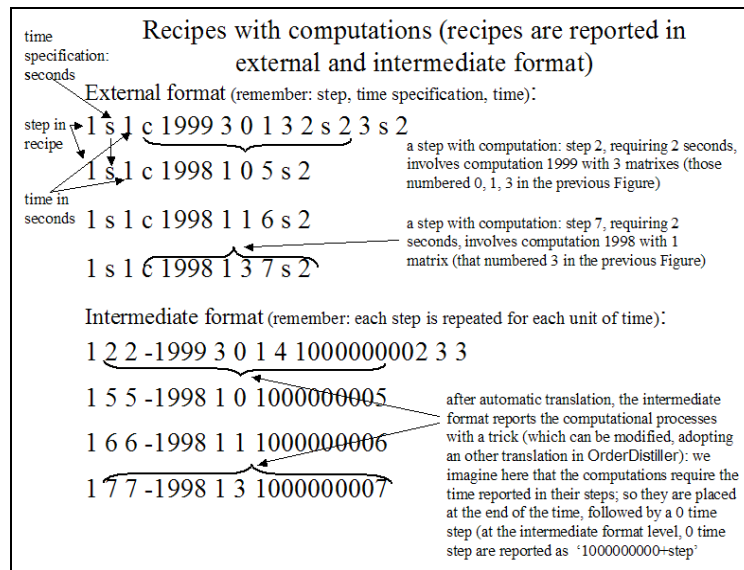


Figura 5. Il formato dei processi computazionali.

Le ricette (vedere figure 5 e 6) possono anche contenere passi computazionali, vale a dire richiami a funzioni espresse sotto forma di metodi Java, con la possibilità di scambiare informazioni tramite matrici comuni di dati.

The Java Swarm code used by the recipes with the example code c 1998

```

/** computational operations with code -1998 (a code for the checking
 * phase of the program
 *
 * this computational code place a number in position 0,0 of the
 * unique received matrix and set the status to done
 */
public void c1998(){

    mm0=(MemoryMatrix) pendingComputationalSpecificationSet.
        getMemoryMatrixAddress(0);
    layer=pendingComputationalSpecificationSet.
        getOrderLayer();

    mm0.setValue(layer,0,0,1.0);
    mm0.print();

    done=true;
} // end c1998

```

Figura 6. Un esempio di Java code di un passo computazionali

La simulazione, da pura sequenza di eventi, può così trasformarsi in sequenza di eventi e computazioni, con una totale flessibilità e estensibilità a qualsiasi metodologia decisionale.

Tramite i passi computazionali è anche possibile che una ricetta contenga il lancio di un altro ordine fondato su un'altra ricetta che opererà indipendentemente dalla prima, rendendo così asincroni e non rigidamente pre-determinati gli avvenimenti..

... A "CHI FA CHE COSA"

Il secondo formalismo, dopo la descrizione delle azioni, è relativo a "chi fa che cosa"; vale a dire le unità produttive, intese in senso molto ampio.

Simple production unit data are reported in a text file
(unitData/unitBasicData.txt)

unit_#	useWarehouse	prod.phase_#	fixed_costs	variable_costs
1	1	11	12	1
2	1	0	0	0
3	1	3	15	2
4	1	0	0	0
5	1	51	12	2
6	1	6	11	20
7	1	12	23	1
8	1	8	22	11
9	1	13	7	12
10	1	18	40	7
11	1	11	5	1

Figure 7. Descrizione di unità produttive semplici.

Possiamo disporre di unità semplici (figura 7), capaci di svolgere un solo compito, oppure di unità complesse (figure 8 e 9), capaci di svolgere molto compiti, in tal caso anche tenendo conto di costi e tempi necessari per passare da un compito all'altro.

	A	B	C	D	E	F	G	H
1	nOfPhases	ToDealWith						
2								
3	phase 1	fixed costs 1	variable costs 1	inventories in production 1				
4	phase 2	fixed costs 2	variable costs 2	inventories in production 2				
5	...							
6	phase n	fixed costs n	variable costs n	inventories in production n				
7								
8	sc 1 1	sc 1 2	...	sc 1 n				
9	sc 2 1	sc 2 2	...	sc 2 n				
10				
11	sc n 1	sc n 2	...	sc n n				
12								
13	st 1 1	st 1 2	...	st 1 n				
14	st 2 1	st 2 2	...	st 2 n				
15				
16	st n 1	st n 2	...	st n n				
17								
18	Remarks							
19	hopefully, fixed costs are the same for all phases.							
20	anyway, when the unit state is undefined (no production made) we use the fixed costs of the first row							
21								
22	inventory production can be 0 (no) or 1 (yes); normally, only one row is set to 1							
23	if we find more than one row set to 1, the first one is chosen							
24								
25	sc i j = setup costs from state i to state j				st i j = setup time from state i to state j			
26								

Figura 8. Unità complesse: schema generale.

Questa descrizione è compiuta tramite file esterni al simulatore, con convenzioni molto semplici.

	A	B	C	D	E	F	G	H	I	J	K
1	3										
2											
3			201	1	1	0					
4			2001	1	1	0					
5			2	1	1	1					
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											

Figura 9. Un esempio di unità complessa;.

Le unità produttive possono essere contestualizzate sia considerandole congiuntamente all'interno di una organizzazione-impresa, sia includendo più imprese di un unico ambito di distretto o di altro spazio territoriale.

SEQUENZE DI EVENTI

Il terzo formalismo è quello che presiede alla sequenza degli eventi: descrive una sequenza nel tempo, lineare o con cicli ripetitivi, degli ordini (contenenti le ricette) da eseguire.

L'intelligenza del sistema sta negli ordini e nelle ricette, che si muovono all'interno delle unità produttive secondo le proprie indicazioni interne, spostandosi di unità produttive in unità produttiva, con sequenze rigide, o flessibili qualora sia necessario compiere scelte.

SCELTE NELLE UNITÀ

Le scelte da compiere sono di più tipi:

- a) a quale unità, tra quelle che possono svolgere un certo compito, attribuire l'esecuzione di un determinato passo descritto in una ricetta;
- b) quale priorità assegnare ad ogni ricetta nella gestione delle code all'interno di ciascuna unità;
- c) decidere se un'unità accetta o no di eseguire un passo assegnato sulla base di una ricetta (accetta o no di prendere in carico la ricetta per la parte di competenza);
- d) introdurre nuove ricette, con nuovi passi produttivi, e nuove unità, con combinazioni di competenze già esistenti o con nuove competenze.

La gestione dei meccanismi di scelta è fondata sul formalismo ERA della figura 10 (Gilbert e Terna, 2000): gli agenti, per scegliere, interrogano un gestore di regole che può essere a volta a volta un insieme di regole, un algoritmo genetico, un sistema a classificatore, una funzione a rete neurale, oppure – aspetto del tutto innovativo – un agente naturale, vale a dire una persona che interagisce con un modello di simulazione all'interno del quale esiste un agente artificiale che opera come suo rappresentante. L'agente artificiale che rappresenta una persona nel modello è un *avatar* (l'incarnazione di una divinità nella mitologia indù, ma anche termine tecnico informatico).

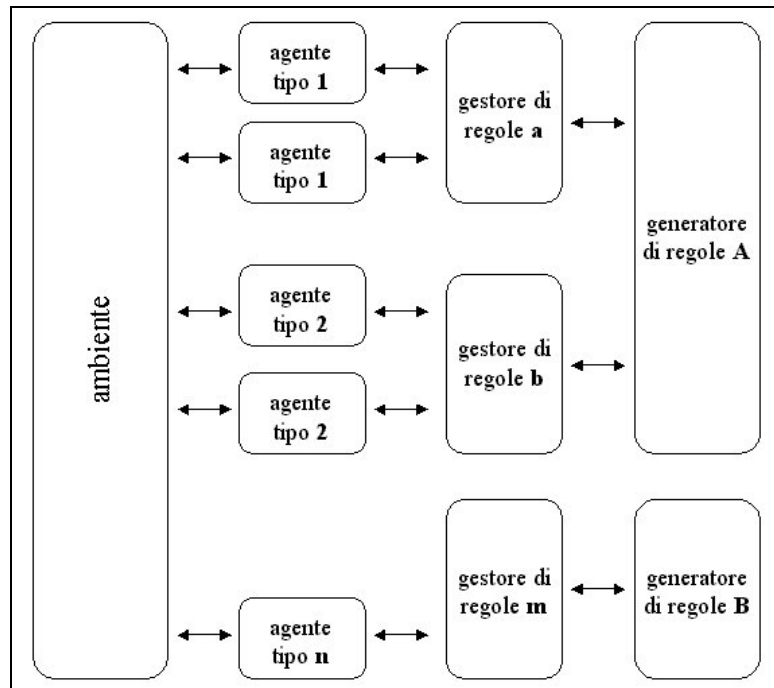


Figura 10 - Schema ERA, *Environment-Rules-Agents*.

Ricette e unità possono produrre un risultato, sotto forma di *output* della produzione, oppure no, per effetto di manchevolezze o incoerenze, come accade nella realtà.

Il rifiuto sub c) a svolgere un'attività provoca altri processi di scelta, sino a che siano esaurite tutte le possibilità (se nessuna riesce la ricetta è abbandonata, contabilizzando gli oneri inutilmente sostenuti sino a quel passo).

L'inclusione della decisione c) nella simulazione è particolarmente interessante se si comprende nel modello anche la gestione del credito: si può allora trattare della decisione di concedere o no credito ad un'altra unità produttiva, con propagazione degli effetti dal lato finanziario a quello reale.

MODALITÀ DI SCELTA

Le scelte possono essere effettuate con regole semplici, oppure in modo casuale (il che rappresenta una "robusta" approssimazione di ciò che accade nella realtà).

In modo più sofisticato, le scelte possono essere simulate tramite la selezione di popolazioni di prove, utilizzando algoritmi genetici; oppure possono essere verificate nel comportamento di agenti naturali chiamati ad interagire con la simulazione.

Utilizzando algoritmi genetici, le decisioni da assumere possono essere frutto di scelte selezionate grazie ai risultati della simulazione stessa, cui si sottopongono non tanto singole decisioni, quanto sequenze di scelte.

L'abbinamento di un algoritmo genetico e di un modello di simulazione è particolarmente fruttuosa in quanto la valutazione dei risultati di ogni popolazione di scelte, sulla base dei quali generare evolutivamente le popolazioni successive, quando si è in presenza di fenomeni complessi non può essere effettuata agevolmente con una funzione espressa analiticamente; la valutazione può invece essere condotta più vantaggiosamente con un processo di calcolo (e cioè con un esercizio di simulazione).

Utilizzando agenti naturali le scelte possono inoltre essere compiute con l'interazione tra simulazione ed agenti naturali, come forma innovativa di esperimento economico. Ciò consente l'uso del simulatore come strumento: (i) di sperimentazione, per comprendere come le persone decidono nelle organizzazioni, secondo l'impostazione metodologica di Simon (1997); (ii) di sussidio alla formazione.

Operando con algoritmi genetici o con agenti naturali per sperimentazione o formazione, le scelte, di tutti i tipi, ma in particolare quelle sub d), possono essere valutate secondo la convenienza di un intorno ristretto o allargato rispetto a chi decide (scelte ottimizzanti: all'interno di una organizzazione o impresa; tra organizzazioni o imprese; di distretto; tra produttori e utilizzatori).

Le scelte sub d) possono anche prefigurare la nascita di nuove imprese, che colmano carenze nelle reti di interazioni, secondo la lettura di Burt (1992).

COME OPERA IL SIMULATORE

Il sistema delle scelte può riguardare in primo luogo l'individuazione dell'unità più adatta cui affidare un compito. Ciò determina valutazioni a volta a volta differenziate a seconda dell'interesse di chi compie la scelta: se le unità sono state contestualizzate all'interno di imprese e queste ultime a loro volta all'interno di distretti o di sistemi è evidente che l'interesse è molto diverso a seconda che la decisione sia assunta all'interno di un'impresa o di un'altra, all'interno del distretto o al di fuori di esso.

Oltre alla scelta della unità tra più possibilità, si pone il problema della costruzione dei sistemi di priorità all'interno delle unità o all'interno delle code a valle delle quali si collocano le decisioni (in termini di formalismo della simulazione si tratterebbe semplicemente di costruire una unità che abbia come solo scopo quello di determinare decisioni).

La gestione delle sequenze di decisione, vale a dire del riordino delle code, si collega direttamente ai meccanismi di simulazione delle decisioni favorendo in modo diretto l'introduzione di algoritmi genetici, o di sistemi a classificatore, che consentano di gestire le sequenze migliori, creando processi di apprendimento che utilizzano a loro volta internamente il simulatore. In altri termini, la simulazione sperimenta più soluzioni, a valle delle quali, scelta la più conveniente, procede con i passi successivi. In

questo modo il simulatore svolge due compiti: di esecuzione di prove per scelte; di inserimento di quelle scelte nella riproduzione della realtà simulata.

PROSPETTIVE DI APPLICAZIONE DELLA SIMULAZIONE DI SITUAZIONI DINAMICHE: LA NASCITA DI NUOVE IMPRESE

La funzionalità di simulazione delle decisioni può essere completata tenendo conto della comparsa di nuove ricette, che ricombinino sequenze di attività già note o che richiedono attività nuove.

Dato che il simulatore ricerca le unità capaci di compiere ogni passo delle ricette svolgendo una “chiamata generale” verso tutte le unità produttive, per verificare chi è in grado di compiere le attività che via via si rendono necessarie, la rilevazione delle chiamate cadute nel vuoto segnala carenze strutturali nel sistema produttivo. Si introduce quindi nel simulatore un meccanismo che, operando in tali casi, possa rappresentare la scelta imprenditoriale di costituzione di una nuova unità produttiva.

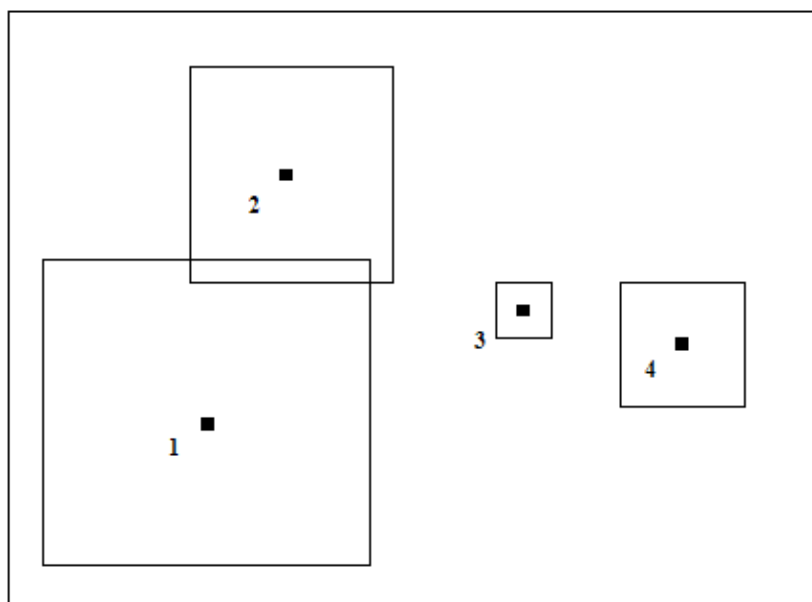


Figura 11. Interazione tra unità produttive e capitale sociale.

E' inoltre utile prevedere che le unità non solo esistano oppure no, ma anche che esibiscano gradi di notorietà diversi (di appartenenza alle reti sociali, nel senso ampio della letteratura sul capitale sociale) per cui una unità esistente, ma non collegata alla rete sociale che stiamo studiando, è come se non appartenesse al sistema produttivo (si veda la figura 11, nella quale le unità 3 e 4 non partecipano alla rete delle unità 1 e 2). La sua inclusione nella rete sociale colma un “buco strutturale”, ancora nel senso di Burt (1992).

In questo modo si è anche implicitamente descritto un “sistema evolutivo”. In termini molto generali si prefigura un sistema di simulazione in cui un insieme di codici, capaci di descrivere attività produttive, genera indipendentemente, da un lato, le “cose da fare”, dall'altro, “chi le fa”, in modo anche non coerente. Ciò è consentito dall'adozione dei

due primi due formalismi di *jES*, che descrivono separatamente le azioni da compiere ed i soggetti realizzatori.

La sequenza degli eventi gestita dal simulatore può condurci a scoprire che alcune unità produttive non possono consegnare ad un successore l'ordine del quale hanno compiuto un passo, perché la maglia produttiva che è stata generata è incompleta, nel senso descritto sopra (vedere anche l'appendice tecnica).

In questo ambiente è quindi interessante simulare la nascita di imprese e verificarne la probabilità di sopravvivenza.

APPENDICE TECNICA SULLA SIMULAZIONE DINAMICA.

Le unità produttive sono generate disposte casualmente su una superficie, che non rappresenta la distanza, ma la rete sociale. Le ricette, spostandosi di unità in unità, "vedono" solo quelle hanno aree di visibilità contigue (figura 11); gli spazi di visibilità si modificano.

Si gestiscono tre "viste" del mondo simulato:

- se un'unità produttiva contiene una coda di prodotti da smaltire, essa è indicata con colore progressivamente più evidente in una prima viste;
- se invece l'unità contiene una coda di ricette parzialmente elaborate, che non possono essere consegnate ad altre unità, l'unità stessa è indicata con colore progressivamente più evidente in una seconda rappresentazione;
- in una terza vista le unità produttive sono indicate con un'area che ne rappresenta la visibilità.

Esiste uno spazio dei codici da cui sono generati sottoinsiemi distinti di unità produttive e di ricette. Un generatore di unità produttive interviene via via creando o eliminando unità, nonché incrementandone o decrementandone lo spazio di visibilità.

Se la modificazione ("impresa") introdotta peggiora lo stato del sistema, essa è eliminata. Se lo migliora, è inclusa stabilmente nella simulazione. L'effetto del cambiamento, da individuare ad esempio in termini di riduzione dei due tipi di code, può essere misurato per tutti; per un sottoinsieme di unità; per una singola unità.

BIBLIOGRAFIA

BURT R.S. (1992), *Structural Holes – The Social Structure of Competition*. Cambridge, MA, Harvard University Press.

GILBERT N., TERNA P. (2000), How to build and use agent-based models in social science. *Mind & Society*, n. 1, pp.57-72.

HOLLAND J. (2003), comunicazione durante la lezione introduttiva del convegno Swarm Fest 2003, 14 aprile, Notre Dame, IN.

SIMON H.A. (1997), *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations*. Simon & Schuster, New York.

TERNA P. (2002), Simulazione ad agenti in contesti di impresa. *Sistemi intelligenti*, 1, XIV, pp.33-51.