

**NIGEL GILBERT, AGENT-BASED MODELS,  
THOUSAND OAKS, CA, SAGE, 2008.**

(SAGE SERIES QUANTITATIVE APPLICATIONS IN THE SOCIAL SCIENCES, N. 153)

Valentina Cisbani

Gli Agent-Based Models (Modelli agent-based) sono forme di simulazione computazionale, molto usate nell'ambito della ricerca nelle discipline scientifiche ma a lungo trascurate dagli studi sociali. A partire dagli anni '90, però, le potenzialità delle simulazioni agent-based sono state riconosciute anche in questo ambito e gli studi di settore si sono moltiplicati notevolmente. Nonostante il nuovo interesse, la carenza di corsi specialistici per l'uso e la costruzione di modelli e la scarsità della letteratura è ancora, a detta dell'autore, un dato evidente.

Nigel Gilbert, professore di Sociologia e Metodologia della ricerca sociale dell'Università del Surrey (UK), è uno dei pionieri degli studi sulle simulazioni realizzate attraverso programmi informatici e applicate alla ricerca sociale; editore di una delle riviste più autorevoli in questo campo, il "Journal of Artificial Societies and Social Simulation", è membro della European Social Simulation Association. Dalla lunga esperienza dell'autore nell'ambito delle simulazioni computazionali applicate alla ricerca scientifica nasce il volume *Agent-Based Models*, pubblicato nel 2008 nelle autorevoli *SAGE Publications Series*. Gilbert propone un approccio pragmatico alla materia, fornendo numerosi esempi di modelli già implementati e spiegazioni tecniche sulla programmazione. In ciascuno dei cinque capitoli che costituiscono il volume, la trattazione teorica è integrata da esempi concreti di ricerche svolte con l'utilizzo di tali modelli e dall'illustrazione molto puntuale della programmazione necessaria per implementarli, fino al commento dettagliato di alcuni esempi di codice sorgente.

A partire da questo forte approccio tecnico-informatico, il testo di Gilbert si pone come un'importante ricerca di metodologie per le scienze sociali, orientata a mostrare come l'utilizzo delle simulazioni computazionali possa fornire nuove prospettive alla sociologia, prospettive avvalorate dalle numerose esperienze concrete di ricerca riportate all'interno del volume, dagli studi sulle *Opinion Dynamics* a quelli sui *Consumers Behaviours*.

In questa ottica, i modelli agent-based si affiancano ai metodi fino ad ora impiegati dai ricercatori per la simulazione dei fenomeni sociali, potenziando e arricchendo alcune direzioni di ricerca ancora poco sviluppate. Il confronto tra i più tradizionali modelli

di simulazione teorico-matematica e il modello agent-based ne evidenzia le caratteristiche, le differenze e i campi di applicazione preferenziale. Una prima differenza evidente tra i due sistemi va ricercata nel fatto che i modelli matematici rappresentano agenti omogenei tra di loro, espressioni di rapporti di tipo numerico astratto. Tale elemento chiama in causa uno degli aspetti innovativi connessi con gli agent-based models e cioè che questi permettono di gestire agenti disomogenei, rappresentativi di determinati target reali e dotati di specificità variabili da un agente, o gruppo di agenti, all'altro. Si tratta evidentemente di un fattore di grande importanza per garantire una rappresentazione efficace di dinamiche sociali ad elevato grado di complessità. A livello informatico, questa caratteristica è connessa con il particolare tipo di linguaggio di programmazione adoperato per costruire i modelli agent-based e ne è una diretta conseguenza. La programmazione degli agenti e degli ambienti, infatti, avviene attraverso linguaggi object-oriented (Java, C++, Visual Basic), che producono oggetti/programmi ciascuno dei quali è dotato di un set di caratteristiche proprie. Ogni agente è programmato come un oggetto separato ed ha un alto livello di modularità per cui è possibile modificare una parte definita del codice senza modificare il resto. Le caratteristiche di ciascun agente sono espresse attraverso l'appartenenza a delle classi: "il passo tra object-orientation e agent-based modeling è breve: si crea una classe per ciascun tipo di agente; si stabiliscono caratteristiche immagazzinate nella *memoria* dell'agente (memory); metodi per la *percezione* dell'ambiente (perception) e *l'azione* (performance), in base a un set di *regole* (policy). Infine occorre programmare un determinato numero di agenti e un sistema che stabilisca per ciascuno di essi il proprio turno per agire" (p. 23).

D'altro canto, gli agent-based models, oltre che rappresentare agenti disomogenei, permettono di riprodurre l'interazione degli agenti tra loro e con l'ambiente. Se da un lato, infatti, approcci teorico-matematici come per esempio il System Dynamics, in cui manca la rappresentazione diretta degli agenti, sono risultati efficaci per studiare il comportamento di ampi gruppi omogenei, quelli agent-based sono chiaramente più efficienti nel rappresentare l'interazione e la specificità.

La capacità di rappresentare l'interazione agente/agente e agente/ambiente costituisce il secondo principale elemento di innovazione che fa delle simulazioni computazionali una

valida risorsa per la ricerca sociale. In questo modo, infatti, gli agent-based models “mettono in scena” dei mondi limitati ma coerenti, in base ad un modello e ad una serie di ipotesi introdotte dal ricercatore. L'autore, quindi, li definisce “metodi *computazionale/informatici* che permettono al ricercatore di creare, analizzare e sottoporre ad *esperimento* dei *modelli* composti da *agenti* che interagiscono in un *ambiente*”.

La rappresentazione dell'interazione differenzia gli agent-based models anche dalla cosiddetta ‘microsimulazione’ in cui, partendo da dati reali registrati in un database, viene simulata la variazione nel tempo di tali indicatori in base a regole esplicite. Il fatto che la microsimulazione parta da dati reali le conferisce una solidità scientifica ma, d'altro canto, la definizione delle regole può presentare notevoli inconvenienti, tra cui la difficoltà di rappresentare fattori imprevisti e ‘irregolari’.

A parte il confronto tra i differenti approcci di simulazione, una seconda lettura degli Agent-based models conduce a definirne i principali elementi costitutivi e cioè *agenti*, *ambienti* e *regole*.

Gli *agenti*, attori del sistema, sono definiti nel glossario tecnico posto a completamento del volume: “A computer program, or part of a program, that can be considered to act autonomously and that represents an individual, organization, nation-state, or other social actor”. Nella prefazione al volume Gilbert ne specifica le caratteristiche di completezza, consistenza e non ambiguità, necessarie perché gli agenti possano essere eseguiti da un computer. Si tratta di caratteristiche che contribuiscono al rigore della ricerca, perché gli agenti devono essere accuratamente pianificati prima di realizzare il modello informatico e devono essere, quindi, il risultato di una riflessione teorica attenta e scientificamente fondata. Essi possono essere programmati separatamente grazie ai linguaggi di programmazione *object-oriented* e, quindi, possono essere rappresentativi di enti/individui/gruppi aventi caratteristiche molto varie. Sono dotati di *percezione*, *performance*, *memoria* e *policy*, e sono accomunati da *autonomia* (ciascun agente agisce in base alle proprie caratteristiche e all'ambiente); *abilità sociale* che permette l'interazione con l'ambiente e con gli altri agenti; capacità di *reazione* agli stimoli esterni e *proattività* in vista di obiettivi. Essi, quindi, percepiscono input provenienti da altri agenti o dall'ambiente (percezione, abilità sociale); ne traggono informazioni che vengono immagazzinate nella memoria e, in base

ad un codice di condotta (policy), determinano una reazione o azione adeguata. Le reazioni, però, possono essere diverse da agente ad agente, in base alle policies di ciascuno, alle esperienze immagazzinate nella memoria e al grado di reattività e proattività programmato.

In questo modo, gli Agenti contano su una “corrispondenza ontologica” con il target rappresentato e, in virtù della loro diversità, sono in grado di rappresentare situazioni complesse. Le tecnologie disponibili permettono di creare, per esempio, agenti dotati di una razionalità *limitata*, quindi con una qualche capacità di rappresentare comportamenti parzialmente irrazionali o imprevedibili e di simulare l'apprendimento sia a livello individuale che collettivo.

L'interazione tra gli agenti avviene sempre all'interno di un mondo, un *ambiente virtuale* in cui è possibile avviare (run) la simulazione ed osservarne gli effetti. Anche l'ambiente, come gli agenti, corrisponde ad un contesto reale e ne può rappresentare le caratteristiche con minore o maggiore fedeltà, a seconda degli obiettivi della ricerca e del grado di astrazione del modello. Di fatto, in campo informatico, la rappresentazione dell'ambiente in cui si muovono gli agenti è stata spesso trascurata, ma tende ad assumere un significato nuovo per lo studio dei fenomeni sociali, in cui l'elemento ambientale riveste una grande rilevanza complessiva. Lo sviluppo tecnologico, inoltre, prospetta la possibilità di creare modelli sempre più aderenti al reale, applicabili in contesti molteplici in cui si renda necessaria la rappresentazione fedele di una zona, per esempio, geografica. Significativamente, a proposito delle prospettive future delle simulazioni agent-based, Gilbert cita le sperimentazioni che stanno tentando di integrare i sistemi GIS (Geographical Information Systems), comunemente utilizzati per i navigatori satellitari, nelle simulazioni. In questo modo sarebbe possibile migliorare sensibilmente la rappresentazione degli ambienti virtuali, permettendo ai programmi/agenti di muoversi all'interno di zone descritte in modo dettagliato e realistico. Sebbene gli studi sulle tecnologie GIS non siano ancora facilmente compatibili, a livello di programmazione e di linguaggio, con i modelli agent-based, lo sviluppo tecnologico, nell'opinione dell'autore, non dovrebbe tardare ancora molto a raggiungere l'obiettivo.

Infine, un terzo elemento che costituisce la struttura concettuale dei modelli agent-based consiste nelle *regole* che permettono agli

agenti di muoversi ed interagire. A livello informatico, esistono due principali sistemi di produzione di regole: quello basato su un *Production system* e gli *Artificial Neural Network* (ANN) ovvero reti neurali artificiali. In entrambi i casi un input esterno viene percepito dall'agente che risponde in base a delle prassi codificate: le caratteristiche di percezione, reazione e proattività associate agli agenti trovano qui la loro applicazione evidente. Inoltre, per l'utilizzo delle simulazioni in campo sociale, è importante programmare e regolare un certo tasso di casualità, la presenza di fattori di disturbo e distorsioni, come nel caso in cui un agente di una simulazione adotti le opinioni di un altro apportandovi delle modifiche (come avviene nelle simulazioni delle *Opinion dynamics*).

A fronte di quanto esposto, è prevedibile che l'utilizzo dei modelli agent-based per la ricerca sociale richieda delle competenze informatiche abbastanza elevate da parte dei ricercatori. Dovendo realizzare un programma informatico complesso, la fase della progettazione e quella del debugging, che precedono e seguono la scrittura del codice vero e proprio, possono essere notevolmente dispendiose dal punto di vista del tempo e sono spesso sottovalutate dagli studiosi.

Gilbert fornisce tuttavia indicazioni su strumenti *ad hoc* che facilitano l'utilizzo dei modelli agent-based dal punto di vista informatico e permettono di ridurre notevolmente il rischio di errori di programmazione. Nati dalla collaborazione di esperti nel settore e disponibili in modalità open source per l'uso non commerciale, questi tool mettono a disposizione parti di codice già pronte e si articolano a seconda della facilità di utilizzo e della flessibilità. Più adattabili e flessibili, ma di uso piuttosto complesso sono invece le Libraries, mentre i 'modeling environments' sono i più adatti per le simulazioni con un basso livello di complessità. La Tabella I riporta una comparazione tra gli strumenti più diffusi, Swarm, Repast, Mason e NetLogo.

A Comparison of Swarm, Repast, Mason and NetLogo

	Swarm	Repast	Mason	NetLogo
License*	GPL	GPL	GPL	Free, but not open source
Documentation	Patchy	Limited	Improving, but limited	Good
User Base	Diminishing	Large	Increasing	Large
Modelling Language(s)	Objective-C, Java	Java, Python	Java	NetLogo
Speed of Execution	Moderate	Fast	Fastest	Moderate
Support for Graphical User Interface Development	Limited	Good	Good	Very easy to create using "point and click"
Built-in Ability to Create Movies and Animations	No	Yes	Yes	Yes
Support for Systematic Experimentation	Some	Yes	Yes	Yes
Ease of Learning and Programming	Swarm	Moderate	Moderate	Good
Ease of Installation	Swarm	Moderate	Moderate	Very easy
Link to Geographical Information System	Swarm	Yes	Yes	No

\*GPL: General Public Licence, <http://www.gnu.org/copyleft/gpl.html>

A conclusione di questo quadro d'insieme, merita un accenno il tema della programmazione di agenti in grado di attivare forme di apprendimento, adattivo o basato sull'interazione. L'argomento viene affrontato nella sezione conclusiva dell'opera e inquadrato tra gli sviluppi futuribili degli agent-based models. Tra queste prospettive, l'autore cita anche complesse sperimentazioni che mirano alla creazione di agenti in grado di apprendere un linguaggio (Simulation language) e di accrescere il proprio vocabolario attraverso l'interazione, attivando forme embrionali di comunicazione. Uno sviluppo importante per gli Agent-Based Models che, rispetto ad altri approcci di simulazione, sono proprio caratterizzati dallo scambio di informazioni tra gli agenti e una tra le sfide che, nell'opinione di Gilbert, sono realisticamente vicine a realizzarsi nell'arco del prossimo decennio.

Table I. Comparazione delle Libraries: Swarm, Repast, Mason, NetLogo (p. 50)