AGENT-BASED MODELS

AB Models are used where the interactions are decentralised, and the autonomous agents make their own decisions (perhaps constrained).

∴ AB models are suitable for interactions which are *bottom-up*, not *top-down.*

∴ social and market interactions, rather than engineering or internal organisational interactions.

Using AB models

In ABM/ACE models, a population of software objects is:

— instantiated, and each agent is given

— certain internal states (e.g., preferences, endowments) and

— rules of behaviour (e.g., seek utility improvements).

Using AB models

In ABM/ACE models, a population of software objects is:

- **— instantiated, and each agent is given**
- **— certain internal states (e.g., preferences, endowments) and**
- **— rules of behaviour (e.g., seek utility improvements).**

The agents are then permitted to interact directly with one another and a macrostructure emerges from these interactions.

Patterns Emerge

Patterns in this macrostructure may then be (Axtell, 2005):

— compared with empirical data,

— to revise agent internal states and rules, and

— the process repeated until an empirically plausible model obtains.

e.g. ACE stock markets have been used to model heterogeneous agents: will the stylised features of such markets emerge?

Patterns Emerge

Patterns in this macrostructure may then be (Axtell, 2005):

— compared with empirical data,

— to revise agent internal states and rules, and

— the process repeated until an empirically plausible model obtains.

e.g. ACE stock markets have been used to model heterogeneous agents: will the stylised features of such markets emerge? Yes.

< >

What is an Agent?

An agent: a self-centred program that controls its own actions based on its perceptions of its operating environment.

Derived from the Distributed AI notion of a network of calculating nodes.

What is an Agent?

An agent: a self-centred program that controls its own actions based on its perceptions of its operating environment.

Derived from the Distributed AI notion of a network of calculating nodes.

Example: the automata in Conway’s Game of Life or Schelling’s Segregation game or the couples in March & Lave’s Sons and Daughters game..

What is an Agent?

An agent: a self-centred program that controls its own actions based on its perceptions of its operating environment.

Derived from the Distributed AI notion of a network of calculating nodes.

Example: the automata in Conway’s Game of Life or Schelling’s Segregation game or the couples in March & Lave’s Sons and Daughters game..

Another example of an agent that won $2,000,000 in a challenge by the U.S. Department of Defense in October 2005 ...

MDV

Agents and agency

Wooldridge & Jennings (1995) would give computer agents these properties:

- **autonomy: no others control their actions and internal state,**
- **social ability: can interact and communicate with other agents**
- **reactive: they perceive their environment and respond**
- **pro-active: they initiate goal-directed actions**
- **(intentionality: metaphors of beliefs, decisions, motives, and even emotions)**

Further agent features:

plus (Epstein 1999):

- heterogeneity: not “representative” but may differ
- local interactions: in a defined space
- boundedly rational (Simon): information, memory, computational capacity
- non-equilibrium dynamics: large-scale transitions, tipping phenomena

Eight Desired Attributes of Modelled Agents (G&T)

1. **Knowledge & beliefs.**
   Agents act based on their knowledge of the environment (including other agents), which may be faulty — their beliefs, not true knowledge.
2.

Eight Desired Attributes of Modelled Agents (G&T)

1. **Knowledge & beliefs.**
   Agents act based on their knowledge of the environment (including other agents), which may be faulty — their beliefs, not true knowledge.
2. **Inference.**
   Given a set of beliefs, an agent might infer more information.
3.

Eight Desired Attributes of Modelled Agents (G&T)

1. **Knowledge & beliefs.**
   Agents act based on their knowledge of the environment (including other agents), which may be faulty — their beliefs, not true knowledge.
2. **Inference.**
   Given a set of beliefs, an agent might infer more information.
3. **Social models.**
   Agents, knowing about interrelationships between other agents, can develop a "social model", or a topology of their environment: who's who. etc.

Eight Desired Attributes ...

4.

Eight Desired Attributes ...

4. **Knowledge representation.**
Agents need a representation of beliefs: e.g. predicate logic, semantic (hierarchical) networks, Bayesian (probabilistic) networks.

> **[Sebastian] Thrun [leader of the winning team in the 2005 DARPA Grand Challenge] had a Zen-like revelation: “A key prerequisite of true intelligence is knowledge of one’s own ignorance,” he thought. Given the inherent unpredictability of the world, robots, like humans, will always make mistakes. So Thrun pioneered what’s known as probabilistic robotics. He programs his machines to adjust their responses to incoming data based on the probability that the data are correct. — Pacella (2005).**

Eight Desired Attributes ...

5.

< >

Eight Desired Attributes ...

5. **Goals.**
   Agents driven by some internal goal, e.g. survival, and its subsidiary goals (food, shelter). Usually definition and management of goals imposed on the agent.

6.

Eight Desired Attributes ...

5. Goals.
   Agents driven by some internal goal, e.g. survival, and its subsidiary goals (food, shelter). Usually definition and management of goals imposed on the agent.
6. Planning.
   Agent must (somehow) determine what actions will attain its goal(s). Some agents modelled without teleology (simple trial-and-error), others with inference (forward-looking), or planning.
7.

Eight Desired Attributes ...

5. **Goals.**
   Agents driven by some internal goal, e.g. survival, and its subsidiary goals (food, shelter). Usually definition and management of goals imposed on the agent.

6. **Planning.**
   Agent must (somehow) determine what actions will attain its goal(s). Some agents modelled without teleology (simple trial-and-error), others with inference (forward-looking), or planning.

7. **Language.**
   For communication (of information, negotiation, threats). Modelling language is difficult. (Want to avoid inadvertent communication, e.g. through the genome of a population in the GA.)

8.

Eight Desired Attributes ...

5. **Goals.**
   Agents driven by some internal goal, e.g. survival, and its subsidiary goals (food, shelter). Usually definition and management of goals imposed on the agent.

6. **Planning.**
   Agent must (somehow) determine what actions will attain its goal(s). Some agents modelled without teleology (simple trial-and-error), others with inference (forward-looking), or planning.

7. **Language.**
   For communication (of information, negotiation, threats). Modelling language is difficult. (Want to avoid inadvertent communication, e.g. through the genome of a population in the GA.)

8. **Emotions.**
   Emergent features? Significant in modelling agents? Or epiphenomenal?

How to Model Agent Architecture?

Early approach to modelling cognitive abilities (symbolic paradigm) was fragile, complex, and lacked common sense.

How to Model Agent Architecture?

Early approach to modelling cognitive abilities (symbolic paradigm) was fragile, complex, and lacked common sense.

Since then, five approaches:

1. **Production Systems**
2. **Object Orientation**
3. **Language Parsing & Generation**
4. **Machine-Learning Techniques, and (most recently)**
5. **Probabilistic Robotics — Stanley (Thrun et al. 2005).**

How to Model Agent Architecture?

Early approach to modelling cognitive abilities (symbolic paradigm) was fragile, complex, and lacked common sense.

Since then, five approaches:

1. **Production Systems**
2. **Object Orientation**
3. **Language Parsing & Generation**
4. **Machine-Learning Techniques, and (most recently)**
5. **Probabilistic Robotics — Stanley (Thrun et al. 2005).**

Ignore 3., 4. last lecture, 5. too new.

Production Systems

Contain:

1. a set of rules (a condition + an action),
2. a working memory, and
3. a rule interpreter (is the condition satisfied? if so, act)

No prespecified order of rules: contingent.

The agent’s designer specifies how to break ties among rules.

Object Orientation

In “object-oriented” programming languages:

- **“objects” are program structures containing data + procedures for operating on those data;**
- **the data are stored in “slots” inside the object;**
- **the procedures are called “methods”;**
- **objects created from templates called “classes”;**
- **classes are ranked in a hierarchy, with subordinate classes more specialised.**

< >

Modelling pedestrian flow.

e.g. Pedestrian flow in a shopping mall —

- **Class: pedestrian;**
- **Slots: location, direction, gait;**
- **Subclass 1: lone walkers;**
- **Subclass 2: group walkers (with a List of Who, and Interactions with others in the group).**

Modelling pedestrian flow.

e.g. Pedestrian flow in a shopping mall —

- Class: pedestrian;
- Slots: location, direction, gait;
- Subclass 1: lone walkers;
- Subclass 2: group walkers (with a List of Who, and Interactions with others in the group).

If the rules are specified at the class level, then all agents share the rules, but with different attributes.

Modelling pedestrian flow.

e.g. Pedestrian flow in a shopping mall —

- Class: pedestrian;
- Slots: location, direction, gait;
- Subclass 1: lone walkers;
- Subclass 2: group walkers (with a List of Who, and Interactions with others in the group).

If the rules are specified at the class level, then all agents share the rules, but with different attributes.

OO computer languages: C++, Lisp, Java. etc.

Probabilistic Robotics

In the 2004 DARPA Grand Challenge, robots used Production System architecture.

Results:

Probabilistic Robotics

In the 2004 DARPA Grand Challenge, robots used Production System architecture.

Results: The most successful entrant in the 2004 race completed just 7.4 miles of the 150-mile off-road (desert) course, and only six of the fifteen cars competing travelled even 1.3 miles.

Probabilistic Robotics

In the 2004 DARPA Grand Challenge, robots used Production System architecture.

Results: The most successful entrant in the 2004 race completed just 7.4 miles of the 150-mile off-road (desert) course, and only six of the fifteen cars competing travelled even 1.3 miles.

In the 2005 Grand Challenge, many robots used probabilistic (or Bayesian or fuzzy-logic) architecture.

Results:

Probabilistic Robotics

In the 2004 DARPA Grand Challenge, robots used Production System architecture.

Results: The most successful entrant in the 2004 race completed just 7.4 miles of the 150-mile off-road (desert) course, and only six of the fifteen cars competing travelled even 1.3 miles.

In the 2005 Grand Challenge, many robots used probabilistic (or Bayesian or fuzzy-logic) architecture.

Results: “Stanley,” Stanford’s robotic Volkswagen Touareg beat the field, completing the 132-mile race with a winning time of 6 hours 53 minutes and 58 seconds (an average speed of 19.1 mph). Four other vehicles successfully completed the race. All but one of the 23 finalists in the 2005 race surpassed the 7.36 mile distance completed by the best vehicle in the 2004 race.

< >

Grand Challenge Rules

- The vehicle must travel autonomously on the ground in under ten hours.
- The vehicle must stay within the course boundaries as defined by a data file provided by DARPA.
- The vehicle may use GPS and other public signals.
- No control commands may be sent to the vehicle while en route.
- The vehicle must not intentionally touch any other competing vehicle.
- An autonomous service station is permitted at a checkpoint area approximately halfway between start and finish.

The Stanford team won the first prize of US $2,000,000 in 2005.

Modelling the Environment

Definition of the environment depends on what is being modelled.

For individuals:

- **move in a space, or on a network;**
- **use sensors to perceive the environment, including other agents;**
- **perhaps be able to affect the environment directly;**
- **perhaps receive and send signals in the environment.**

For computer agents, the order of agents running can be crucial (“concurrency”). Sometimes, buffering their signals is sufficient.

< >

G & T Use NetLogo to build a multi-agent simulation.

Economic Journal June 2005 Feature —

- focussed on Complex Adaptive Systems CAS in economics
- appeared just after Leombruni & Richiardi asked, “Why are economists sceptical about agent-based simulations?” (*Physica A* 355: 103–109, 2005.)
- included 4 papers: introduced by Markose, with papers by Axtell, Robson, and Durlauf
- addressing, respectively,
  - — markets as complex adaptive systems,
  - — formal complexity issues,
  - — the co-evolutionary Red Queen effect and novelty, and
  - — the empirical and testable manifestations of CAS in economic phenomena.

Markose and the *EJ* Feature on CAS:

- **many “anomalies” not understood or modelled using conventional optimisation economics:**
  - **— innovation,**
  - **— competitive co-evolution,**
  - **— persistent heterogeneity,**
  - **— increasing returns,**
  - **— “the error-driven processes behind market equilibrium,”**
  - **— herding,**
  - **— crashes and extreme events such as October 1987.**
- **need the “adaptive or emergent methods” of ACE simulation**

< >

Moreover ...

Axtell (2005) argues that:

- **the decentralised market as a whole can be seen as a collective computing device**
- **the parallel distributed agent-based models of *k*-lateral exchange → the specific level of complexity (polynomial) in calculations of equilibrium prices and allocations.**

Simon’s Bounded Rationality

Agent-based models, following Simon (1982), also assume Bounded Rationality. Indeed, in the absence of Turing machine (universal calculator), it is difficult not to.

But Epstein (2006) reflects:
“One wonders how the core concerns and history of economics would have developed if, instead of being inspired by continuum physics ... blissfully unconcerned as it is with effective computability — it had been founded on Turing. Finitistic issues of computability, learnability, attainment of equilibrium (rather than mere existence), problem complexity, and undecidability, would then have been central from the start. Their foundational importance is only now being recognized.

Epstein on the virtues of boundedly rational agents ...

“As Duncan Foley summarizes:

> `The theory of computability and computational complexity suggest that there are two inherent limitations to the rational choice paradigm.
>
> One limitation stems from the possibility that the agent’s problem is in fact undecidable, so that no computational procedure exists which for all inputs will give her the needed answer in finite time.
>
> A second limitation is posed by computational complexity in that even if her problem is decidable, the computational cost of solving it may in many situations be so large as to overwhelm any possible gains from the optimal choice of action.’ (See Albin 1998, 46).”

ABM → Generative Explanation:

Generative explanation (Epstein 2006):

“If you haven’t grown it, you haven’t explained its emergence.”

To answer: how could the autonomous, local interactions of heterogeneous boundely rational agents generate the observed regularity (that emerges)?

— Generative sufficiency is a necessary but not sufficient condition for explanation. Each realisation is a strict deduction.

Grüne-Yanoff (2006) argues to distinguish *functional explanations* (easier for simulators) from *causal explanations* (much less achievable for social scientists).

Truth and Beauty

Epstein (2006): does AB simulation lack beauty?

Russell: Mathematics as cold, austere, supreme beauty.

Russell: Beauty when “the premises achive more than would have been thought possible, by means which appear natural and inevitable.”

The first damns computer simulation, but the second can occur with emergence from AB models.

Epstein compares different schools of classical music: German v. French.

Truth (from agent-based modelling) can be beautiful too.

Formalisation of Agent-Based Models

Epstein (2006): every agent model is a computer program.

∴ Turing computable

But for every Turing machine, ∃ a unique corresponding and equivalent

partial recursive function.

They might be extremely complex and difficult to interpret, but they exist.

Hence: ”recursive” or “effectively computable” or “constructive” or “generative” (after Chomsky) social science.

Validation of Agent-Based Models

Moss & Edmonds (2005): for AB models at least two stages of empirical validation.

1. **the micro-validation of the behaviour of the individual agents in the model, by reference to data on individual behaviour.**

2.

Validation of Agent-Based Models

Moss & Edmonds (2005): for AB models at least two stages of empirical validation.

1. **the micro-validation of the behaviour of the individual agents in the model, by reference to data on individual behaviour.**
2. **macrovalidation of the model’s aggregate or emergent behaviour when individual agents interact, by reference to aggregate time series.**

   with the emergence of novel behaviour, possible surprise and possible highly non-standard behaviour, difficult to verify using standard statistical methods.

   ∴ only qualitative validation judgments might be possible.

Simulation and Necessity?

Mathematical “model A” comprises the conjunction ($a_1 \wedge a_2 \wedge a_3 \cdots \wedge a_n$), where $\wedge$ means “AND”, and the a_i denote the elements (equations, parameters, initial conditions, etc) that constitute the model.

Simulation and Necessity?

Mathematical "model A" comprises the conjunction ($a_1 \wedge a_2 \wedge a_3 \cdots \wedge a_n$), where $\wedge$ means "AND", and the a_i denote the elements (equations, parameters, initial conditions, etc) that constitute the model.

***Sufficiency*: If model A exhibits the desired target behaviour B, then model A is sufficient to obtain exhibited behaviour B: $A \Rightarrow B$**

Thus, any model that exhibits the desired behaviour is sufficient, and demonstrates one conjunction of conditions (or model) under which the behaviour can be simulated.

Simulation and Necessity?

Mathematical "model A" comprises the conjunction ($a_1 \wedge a_2 \wedge a_3 \cdots \wedge a_n$), where $\wedge$ means "AND", and the a_i denote the elements (equations, parameters, initial conditions, etc) that constitute the model.

***Sufficiency*: If model A exhibits the desired target behaviour B, then model A is sufficient to obtain exhibited behaviour B: $A \Rightarrow B$**

Thus, any model that exhibits the desired behaviour is sufficient, and demonstrates one conjunction of conditions (or model) under which the behaviour can be simulated.

But if there are several such models, how can we choose among them? And what is the set of all such conjunctions (models)?

Necessity

Necessity: **Only those models *A* belonging to the set of necessary models $\mathcal{N}$ exhibit target behaviour *B*.**

That is, ($A \in \mathcal{N}$) $\Rightarrow$ *B*, and ($D \notin \mathcal{N}$) $\not\Rightarrow$ *B*.

Necessity

Necessity: **Only those models *A* belonging to the set of necessary models $\mathcal{N}$ exhibit target behaviour *B*.**

That is, $(A \in \mathcal{N}) \Rightarrow B$, and $(D \notin \mathcal{N}) \nRightarrow B$.

A difficult challenge: determine the set of necessary models, $\mathcal{N}$.

Since each model $A = (a_1 \wedge a_2 \wedge a_3 \cdots \wedge a_n)$, searching for the set $\mathcal{N}$ of necessary models means searching in a high-dimensional space, with no guarantee of continuity, and a possible large number of non-linear interactions among elements.

Lack of Necessity Means ...

For instance, if $D \not\Rightarrow B$, it does not mean that all elements a_i of model D are invalid or wrong, only their conjunction, that is, model D.

It might be only a single element that precludes model D exhibiting behaviour B.

But determining whether this is so and which is the offending element is a costly exercise, in general, for the simulator.

Without clear knowledge of the boundaries of the set of necessary models, it is difficult to generalise from simulations.

Lack of Necessity Means ...

For instance, if $D \nRightarrow B$, it does not mean that all elements a_i of model D are invalid or wrong, only their conjunction, that is, model D.

It might be only a single element that precludes model D exhibiting behaviour B.

But determining whether this is so and which is the offending element is a costly exercise, in general, for the simulator.

Without clear knowledge of the boundaries of the set of necessary models, it is difficult to generalise from simulations.

Only when the set $\mathcal{N}$ of necessary models is known to be small (such as in the case of DNA structure by the time Watson & Crick were searching for it) is it relatively easy to use simulation to derive necessity.

Formalisation of Validation

Let set *P* be the possible range of observed outputs of the real-world system.

Let set *M* be the exhibited outputs of the model in any week.

Let set *S* be the specific, historical output of the real-world system in any week.

Let set *Q* be the intersection, if any, between the set *M* and the set *S*, $Q \equiv M \cap S$.

We can characterise the model output in several cases. (Mankin et al. 1977).

Five Cases for Validation

a. **no intersection between *M* and *S* (*Q* = ∅), then the model is *useless.***

b. **intersection *Q* is not null, then the model is *useful,* to some degree: will correctly exhibit some real-world system behaviours, will not exhibit other behaviours, and will exhibit some behaviours that do not historically occur. Both incomplete and inaccurate.**

c. **If *M* is a proper subset of *S* ($M \subset S$), then all the model’s behaviours are correct (match historical behaviours), but the model doesn’t exhibit all behaviour that historically occurs: accurate but *incomplete.***

d. **If *S* is a proper subset of *M* ($S \subset M$), then all historical behaviour is exhibited, but will exhibit some behaviours that do not historically occur: complete but *inaccurate.***

e. **If the set *M* is equivalent to the set *S* ($M \Leftrightarrow S$), then (in your dreams!) the model is complete and accurate.**

Validation Relationships

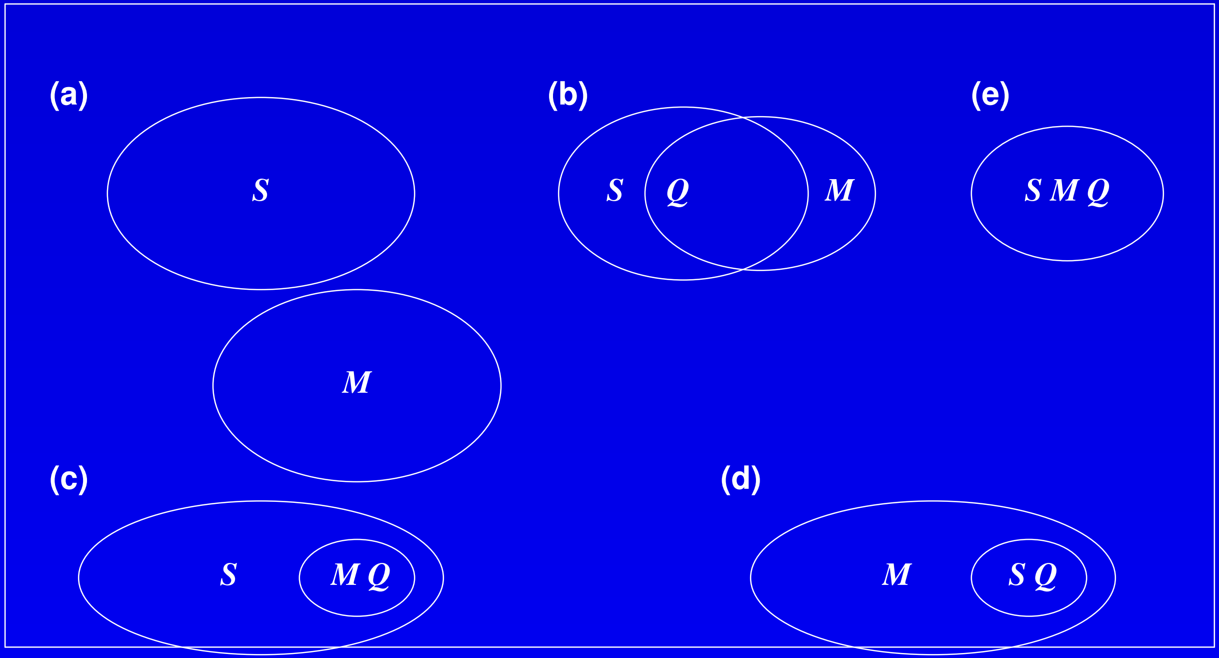

Figure 2: Validity relationships (after Haefner (2005)).

Modelling Goals

One goal: to construct and calibrate the model so that $M \approx Q \approx S$: there are very few historically observed behaviours that the model does not exhibit, and there are very few exhibited behaviours that do not occur historically.

The model is close to being both complete and accurate.

Modelling Goals

One goal: to construct and calibrate the model so that $M \approx Q \approx S$: there are very few historically observed behaviours that the model does not exhibit, and there are very few exhibited behaviours that do not occur historically.

The model is close to being both complete and accurate.

In practice, a modeller might be happier to achieve case d., where the model is complete (and hence provides sufficiency for all observed historical phenomena), but not accurate.

Measures of Validity

A measure of validity which balances the Type I error of inaccuracy with the Type II error of incompleteness.

Define a metric *m*() (a ratio scale) on the sets.

Define *inaccuracy* α as

$$\alpha \equiv 1 - \frac{m(Q)}{m(M)}\,, \tag{1}$$

and *incompleteness* γ as

$$\gamma \equiv 1 - \frac{m(Q)}{m(S)}\,. \tag{2}$$

Continued ...

A measure of degree of validation V: a weighted average of inaccuracy α and incompleteness γ:

$$V \equiv v(1-\alpha) + (1-v)(1-\gamma) \tag{3}$$

$$\therefore\ V = v\,\frac{m(Q)}{m(M)} + (1-v)\,\frac{m(Q)}{m(S)}$$

$$\therefore\ V = m(Q)\left(\frac{v}{m(M)} + \frac{1-v}{m(S)}\right) \tag{4}$$

The value of the weight v, $0 \le v \le 1$, reflects the tradeoff between accuracy and completeness.

Trade-offs

Possible to reduce incompleteness by generalising the model and so expanding the domain of set *M* until *S* is a proper subset of *M*, as in case d.

Or by narrowing the scope of the historical behaviour to be modelled, so reducing the domain of *S*.

< >

Trade-offs

Possible to reduce incompleteness by generalising the model and so expanding the domain of set *M* until *S* is a proper subset of *M*, as in case d.

Or by narrowing the scope of the historical behaviour to be modelled, so reducing the domain of *S*.

Also be possible to reduce inaccuracy by restricting the model through use of narrower assumptions and so contracting the domain of *M*.

If *M* is sufficiently small to be a proper subset of *S*, as in case c., then the model will never exhibit anhistorical behaviour.

Trade-offs

Possible to reduce incompleteness by generalising the model and so expanding the domain of set *M* until *S* is a proper subset of *M*, as in case d.

Or by narrowing the scope of the historical behaviour to be modelled, so reducing the domain of *S*.

Also be possible to reduce inaccuracy by restricting the model through use of narrower assumptions and so contracting the domain of *M*.

If *M* is sufficiently small to be a proper subset of *S*, as in case c., then the model will never exhibit anhistorical behaviour.

But not guaranteed to maintain a non-null intersection *Q*, and it is possible that the process results in case a., with no intersection.

Look in the Right Place

Reminiscent of the economist looking for his lost car keys under the street light (*M*), instead of near the car where he dropped them in the dark (*S*).

Advocates of simulated solutions, such as Judd (2006), have argued that it is better to “have an approximate answer to the right question, than an exact answer to the wrong question,” to quote Tukey (1962).

References:

- P.S. Albin, ed. 1998. *Barriers and Bounds to Rationality: Essays on Economic Complexity and Dynamics in Interactive Systems*. PUP, Princeton. Introduction by D.K. Foley.
- R. Axtell, The complexity of exchange, *The Economic Journal,* 115 (June), F193–F210, 2005.
- S. Durlauf, Complexity and empirical economics, *The Economic Journal*, 115 (June), F225–F243, 2005.
- J.M. Epstein, Agent-based computational models and generative social science, *Complexity*, 1999.
- J.M. Epstein, Remarks on the Foundations of Agent-Based Generative Social Science, *Handbook of Computational Economics, Volume 2: Agent-Based Modeling*, edited by Leigh Tesfatsion and Kenneth L. Judd, Amsterdam: Elsevier Science, 2006.
- T. Grüne-Yanoff, The Explanatory Potential of Artificial Societies, *Models and Simulations* (Paris, 2006).
- A. Hailu & S. Schilizzi, Are Auctions More Efficient Than Fixed Price Schemes When Bidders Learn? *Australian Journal of Management*, 29(2): 147–168, December 2004.
- R. Leombruni & M. Richiardi, Why are economists sceptical about agent-based simulations? *Physica A* 355: 103–109, 2005.
- S.M. Markose, Computability and evolutionary complexity: markets as complex adaptive systems (CAS), *The Economic Journal*, 115 (June), F159–F192, 2005.
- R.E. Marks (2007), Coffee, segregation, energy, and the law: validating simulation models, *Computational Economics*, forthcoming.
- D.F. Midgley, Marks R.E., and Kunchamwar D. (2007) The Building and Assurance of Agent-Based Models: An Example and Challenge to the Field, *Journal of Business Research*, Special Issue: Complexities in Markets, forthcoming.
- R.M. Pacella (2005), DARPA Grand Challenge, *Popular Science*, October.
- A.J. Robson, Complex evolutionary systems and the Red Queen, *The Economic Journal,* 115 (June), F211–F224, 2005.

- **Simon, H.A. 1982. *Models of Bounded Rationality*. Cambridge, MA: MIT Press.**
- **Simon, H.A. 1987. Rationality in Psychology and Economics. *Rational Choice: The Contrast Between Economics and Psychology*, Hogarth, R.M. and Reder, M.W. eds. Chicago: University of Chicago Press.**
- **S. Thrun, W. Burgard, D. Fox, *Probabilistic Robotics*, MIT Press, September 2005.**
- **M. Wooldridge & N. R. Jennings, Intelligent Agents: Theory and Practice, *Knowledge Engineering Review,* 1995.**

<