AGENT-BASED MARKET DESIGN

Robert Marks
Australian Graduate School of Management[1]
UNSW, Sydney 2052
Australia

ABSTRACT: New kinds of markets demand new kinds of market design. In the past twenty years several new kinds of market have been devised and put into operation, sometimes after several false starts. Designing markets is a new activity for economists, who haven't readily thought of themselves as engineers. Emergence of the market engineer has been hastened by the interest of computer scientists in designing on-line markets, and there has been a three-way marriage of game theory, experimental results, and computer science in the use of computer simulation models to analyse, and to design, new markets. This paper argues for several things. For simulation as an alternative to closed-form analysis in market analysis and design. For agent-based computational economic models as specific simulation models. For explicit validation of such models, using several heuristics. The paper discusses recent research into market design and simulation.

---

1. Email: `bobm@agsm.edu.au`. An earlier version of this paper was presented at the workshop on Agent-Based Models for Economic Policy Research (ACEPOL05), Zentrum für interdisziplinäre Forschung, Bielefeld University, June 30–July 2, 2005.

"... it is typical of many kinds of design problems that the inner system consists of components whose fundamental laws of behavior ... are well known. The difficulty of the design problem often resides in predicting how an assemblage of such components will behave." — Simon (1981).

1. Introduction

By "market design" I mean designing the structure and rules of engagement among economic actors in markets to some specification of performance. By performance I mean the aggregate impacts of the individual decisions of buyers and sellers in the market, which might also have a time dimension. It could be known as the engineering of markets. Many designed markets have been repeated auctions.

For almost as long as recorded history, human societies have used markets and money (as a clear advance over barter) to allocate goods and services. Since Adam Smith, we have begun to understand just what an amazing phenomenon these emergent markets are. With this understanding have arisen demands for new kinds of markets to be designed and implemented, including, most interestingly, some on-line markets with automated buyers and sellers. The marriage between computer scientists and economists is proving very fertile in the new field of market design.

Section 2 argues that there are severe difficulties in designing markets directly, as an optimization, using the analogy of the complexities encountered by software engineers attempting to derive desired software systems. Section 3 lists the kinds of new markets that have been designed, sometimes from scratch, in order to accomplish their tasks of allocation. Section 4 discusses analysis and the use of computer simulation[2] to achieve it. Section 5 discusses modeling learning, including Genetic Algorithms and Reinforcement Learning, both inductive. Section 6 discusses design by iterative analysis, in the presence of syntactic complexity. Section 7 develops a framework for market design. Section 8 focuses on the general problem of market design. Section 9 argues for ACE-based market design, from the bottom up. Section 10 examines barriers to adoption of ACE methods in economics, and focuses on validation of simulation models. Section 11 concludes.

2. The Elemental Complexity of Design

Edmonds & Bryson (2004) examine the shortcomings of formal methods of software design, with some lessons for market design. In formal methods, the specifications of the software are written in a formal language, often of a logical or set-theoretic nature. The specification is thus unambiguous, and the specifications can themselves be formally manipulated.

To achieve correct implementation, there are two major questions that need to be

---

2. The reader should not mistake computer simulation of models of the market with the stochastic sampling that is sometimes called "Monte Carlo simulation." As Judd (1998) points out, the latter is sensitivity analysis of parameter values, often initial conditions; it might be used as part of model simulations, but despite its name it should not be mistaken for them.

answered, which are very similar to the questions that market designers need to answer. The first Edmonds & Bryson call the "programming problem:" how is a given specification translated into a system of software? For market designers, the analogue is: how to map from the specifications of the desired behavior of the market to its structure (rules and environment)? The second question is the "checking problem:" checking whether a specific software system satisfies the formal specification. The market design analogue is: does a given specific market structure (rules and environment) satisfy the specifications?

Edmonds & Bryson (2004) characterise the "formal specification strategy" of software design as a threefold process: first, decide the goals for the software system, second, write a formal specification to meet these goals, and, third, implement a software system that meets these goals. An example of a high-level goal might be "to be responsive to the changing demands of web browsers."

The software engineers need to ensure, first, that the goals are correct (meet the client's needs), second, that the formal specification (which might be expressed in a logical or set-theoretic language) meets the goals, and, third, that the system operates to spec. These are difficult for two broad reasons: first, the programming problem of translating specifications into a program or programs, and, second, the checking problem.

Imagine that there is a program that can translate the formal specification into a software system. Edmonds & Bryson (2004, p.938–9) show that even if such a translation program exists, there is no effective way of producing such a computation that takes us from specifications to a program (or software system) that satisfies the specs. Their proof uses the undecidable *Entscheidungsproblem* (the "halting problem," Turing 1936), and relies on the formal specification being written in a language that is sufficiently "expressive" to formulate an enumeration of formal specifications, together with Gödel's incompleteness proof (Gödel 1931).

Second, even though the checking problem is apparently less ambitious than the programming problem, using similar methods, Edmonds & Bryson (2004, p.939) prove that there is no effective or systematic way of checking whether the program corresponds to the formal specification.

These results are of interest to the market designer, especially if moving from higher-level goals (as determined by the client), to a specification of the market's and traders' behavior, and finally the rules and environment of the market in which human actors buy and sell, can be thought of as analogous to the software design problem of Edmonds & Bryson (2004). Indeed, if it's impossible with well behaved programs to implement the goals and specifications, how much harder with human actors and all their inconsistencies?

Edmonds & Bryson characterise the "programming problem" as resulting in *syntactic complexity*: there is no easy way to predict the resulting behavior of an implementation from its initial set-up. In evolutionary biology, the mapping from genome (structure) to phenome (behavior) exhibits syntactic complexity, and the market design problem we face — mapping from specification of market structure (rules and environment) to market behavior — is similarly syntactically complex.

Given the "messiness" of software systems having to mesh with quite distinct software systems or with direct human interactions, and given the issues of formal specification outlined above, Edmonds & Bryson argue that software engineers need to apply classical scientific experimental methods in order to validate their software systems: to answer the checking problem.

The particular software systems that Edmonds & Bryson are considering are multi-agent systems (MAS), which here are equivalent to agent-based computational economic (ACE) systems. That is, Edmonds & Bryson are arguing for the necessity of using simulation experiments in ensuring that ACE software systems perform correctly (are internally validated, or verified). We argue here that, in the absence of any easy way to map from goals of the market to specification of the market to the market's design because of syntactic complexity and the programming problem, use of ACE systems can allow resolution of the checking problem, and used iteratively can allow new markets to be designed.

The following table contrasts the two approaches of the software engineer and the market designer:

Goals:	for the software	for the market
Specifications:	for the software	for the market performance
Design:	of the MAS software system	of the market (structure and rules)

The irony is that Edmonds & Bryson are designing a multi-agent system of software, while we argue that using agent-based software provides a way of moving from specification to design of markets.[3]

3. New "Designer" Markets

Market design is a discipline that has arisen with the demands for new kinds of markets, and the designing and implementing of new, "designer" markets. We can list five or six kinds of designer markets that have been devised in the last twenty years:

1. Markets for new financial instruments, such as options and derivatives.
   Markets for new financial derivatives were created and traded after Black, Scholes, and Merton solved the 70-year-old problem of pricing options. Previously, financial traders knew that options were valuable, but not how to value them exactly. More recently, there has been research into the rules and micro-structure of stock markets, continuous double-auction trading, through the use of simulated markets (LeBaron (2006))

3. A recent article argues that markets are in effect computational devices themselves: in a paper that draws out the connections between economic exchange, usually in markets, and distributed computation, such as computer networks, Axtell (2005) notes that "bilateral trade produces a kind of social computer which endogenously decentralises economic computations."

2. Markets for emissions trading: for SO_2, CO_2, NO_x.
   Realization that the emissions from anthropogenic processes were, at least potentially, altering the biosphere for the worse was followed only after a lag with the awareness by policy makers that market mechanisms could be harnessed to control such emissions, generally more efficiently than could other mechanisms. See Hailu and Schilizzi (2004) and Janssen and Ostrom (2006).

3. Electro-magnetic spectrum auctions.
   Simultaneous ascending-bid auctions have recently been designed for selling bands of local electro-magnetic spectrum. Use of auctions to choose the new owners and to value these assets slowly replaced so-called "beauty contests," in which subject to certain technical requirements licenses were virtually given away. But these new auction mechanisms at first did not allow for the complementary nature of bands in different localities. Only after intensive efforts by economists advising governments and bidding companies did the successful "3G" auctions occur (Roth 2002, Milgrom 2004).

4. Markets for electricity.
   There has been a move away from centralized engineering-dominated means of allocating electricity load across generators and distribution networks to using market mechanisms of various kinds. Since electricity cannot (easily or cheaply) be stored, previously existing market mechanisms were not appropriate. Instead, several types of new market mechanisms have been introduced. Marks (2006) surveys the origins and development of this application of ACE-based market design.

5. On-line markets and markets for e-commerce.
   With the growth of the use and extent of the Internet over the past ten years, and the dot-com boom, with buying and selling on-line, opportunities for designing on-line markets de novo, as opposed to trying to emulate existing face-to-face markets, have arisen. In the last few years these opportunities have given rise to much work by computer scientists, as well as economists. Indeed, there is a productive research intersection of the two disciplines, as revealed in some of the papers discussed below. See MacKie-Mason and Wellman (2005)

6. Labor clearinghouses.
   Roth (2002) describes earlier work of designing the entry-level labor market (a labor clearinghouse) through which American doctors get their first jobs, and other matching markets.

7. Contract design.
   Not strictly a market, but the negotiations leading to an agreed contract can also be designed. Contract design is another area where agent-based modeling might be used, but negotiation and design of contracts by use of computer simulation and agent-based modeling is only now emerging from its infancy (Jennings et al. 2001).

4. Analysis and Simulation

In order to change markets it is necessary to understand their operation. Before design must come analysis. Once we understand through analysis how the elements of the market of concern work together, we can ask the question of how to improve its operation: how better to design it.

Since Samuelson, economists have sought closed-form solutions to understand the performance of markets. In this approach, economic actors are assumed to be perfectly rational, with the means to solve for equilibria outcomes in complex situations. Economists have examined the existence, uniqueness, and stability of equilibria of economic interactions. When the interactions among economic actors are strategic, the equilibria become Nash equilibria.

But in an operating, real-time actual market, we are not interested just in equilibrium characterization: continual shocks might never allow the system to approach, let alone reach, the equilibrium. Moreover, in a repeated interaction almost any individually rational outcome for each player can be supported as an equilibrium (the Folk Theorem of repeated games). This is particularly so for interactions which have the general character of the iterated Prisoner's Dilemma (IPD).

This paper argues that simulation must be used when closed-form solutions fail, or give an exact answer to the wrong problem, as sometimes happens (Judd 2006). There are four reasons why simulation is the market designer's friend:

1. Tractability: despite improvements in mathematical techniques, it is still very difficult to obtain solutions to the design of some markets, such as continuous double auctions (CDAs).

2. Market designers must characterize out-of-equilibrium behavior, and especially the dynamic behavior of an operating market with fluctuating demand, and perhaps varying numbers of sellers, with unpredictable, varying costs.

3. The assumption of perfect rationality and unlimited computational ability on the part of human traders is unrealistic, and not borne out by laboratory experiments with human subjects. Instead, using computer models of trading agents, designers have modeled economic actors in markets as "boundedly rational" — bounded computational ability, or bounded memory, or bounded perception (Marks 1998).

   Conlisk (1996) gives four reasons for using bounded rationality in economic models: first, evidence of limits to human cognition, second, successful performance of economic models with bounded rationality, third, sometimes unconvincing arguments in favor of unbounded rationality, and, last but not least, the costs of deliberation.

4. To model learning: There are two reasons to include learning in any models used to design markets: first, individuals and organizations learn: a model without learning is not as realistic as one incorporating learning. Bunn & Oliveira (2003) note that many researchers (including Erev & Roth 1998) have shown that learning models predict people's behavior better than do Nash equilibria.

   Moreover, learning can help to eliminate many otherwise legitimate Nash equilibria. Indeed, evolutionary (or learning) game theory has been seen as a solution to the multiplicity of Nash equilibria that occur in closed-form game-theoretic solutions: a priori, all are possible, but to see which are likely in reality, see how players learn and choose amongst them.

5. Learning

Learning can be defined as "Any change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population" — Simon (1983)

There are several sorts of models of learning that could be used in computer simulation models:

1. Artificial Neural Nets (ANNs): derived from machine learning and from biological simplifications of the brain's operation,
2. Evolutionary models, such as Genetic Algorithms (GAs) and Genetic Programming (GP), derived from natural evolution, and
3. Models from psychology experiments, such as Reinforcement Learning (RL).

Here we focus on the GA and RL models, since these have received most use in ACE simulations, although others have been mooted (Duffy 2006).

Among the engineers who pioneered the applications of the GA, it has largely used as an optimizer: asking what is the best value (or fittest individual) in the population. But focusing on the value of the best individual throws away the population's emerging characteristics as a population. It ignores the aggregate level of emerging phenomena. In contrast, Axelrod (1987) not only sought the best-performing strategy in the IPD, but also asked questions at the aggregate level of the population, such as its stability against invasion by a different strategy: e.g. Tit for Tat's stability against Alway Defect. Marks (1989*b*) also examined stability of a population to invasion.

Early use of the GA in both engineering and the social sciences used single-population implementations of the GA. Is this appropriate for simulating social phenomena?

Vriend (2000) distinguished the *social learning* that occurs, for instance, with a single-population GA from *individual learning* that occurs when each agent is modeled as a separate GA. Social learning occurs at the genotypic level: sexual reproduction means that parents can communicate (share information) with their offspring via crossover: therefore over generations, fitter genes or traits can spread through a population covertly, by inheritance of genetic material. In contrast, individual learning occurs only through arm's-length competition, and the selection of fitter individuals as future parents, not through inheritance of genetic material from other player's parents.

When all agents are identical, then modeling them as identical members of a single population is appropriate, and genetic inheritance is not a problem, since the aim might be only to seek the fittest individual, or to allow communication among the group.

When the environment in which the GA operates changes, and when such change is due to the behavior of the species' competitors — co-evolution — then sharing of genetic material blurs the distinction between species or agents. For example: If the GA is being used to explore the behavior of sellers in an oligopolistic market, genetic sharing can only model sub-rosa communication across brands. This is illegal under most antitrust regimes, and therefore in general should not occur in the model, lest the results rely on it.

The answer to the question: how many populations? is then as many as there are distinct agents, or distinct species co-evolving. For example: When each seller in an oligopoly has distinct costs, faces distinct demand responses, perhaps with a distinct actions set, then it should be modeled using a distinct population. Perhaps because each GA has an internal population of individuals, and perhaps revealing the predominant uses of the GA as an optimization tool in engineering, there has been a tendency to think of the GA as modeling heterogeneous players. But a single population assumes homogeneity.

Each string in a GA population could be modeling one of two possibilities: either an individual brand (say), which I have argued above is unrealistic in with a single population, or one possible decision of a selection (the population) that the agent could make. This latter makes sense with a population per distinct player. So there are two possibilities for each new generation. It could comprise: either new individual decision makers (brands), or, alternately, new ideas or heuristics belonging to long-lived players. The former corresponds to Vriend's social learning, and the latter to his individual learning. The two contrasting models of learning do not require the GA for their implementation.

There have been criticisms of the use of GAs previously. Chattoe (1998) argues (correctly) that there has been confusion over the role of the GA. Is it an instrument to search a rugged solution space (Kauffman 1995), or a model of firms' decision making and individual behavior? Dawid (1999) argues that the GA is good at modeling the learning of populations of agents. Curzon Price (1997) argues that the GA can be seen as providing a stream of hypothetical actions or strategies, which may or may not be used. Duffy (2006) concludes that empirical evidence exists that GAs are reasonable "models of adaptive learning by populations of heterogeneous agents."

Whichever version (single- or multi-population) of the GA is used, learning in this model is implicit: it occurs at the population level, not at the individual level — it emerges. With RL models, however, learning is explicit. In both cases, learning is inductive (backwards-looking) rather than deductive (forward-looking).

Arthur (1991, 1993) was the first economist to model explicit agent learning, and to calibrate his models using data from human-subject experiments. In his RL model, how an agent chooses to act later is a function of the outcomes it experienced as a result of earlier choices — the Thorndike effect. At first he calibrated individual learning, but with the artificial stock market (Arthur et al. 1997), he became interested in data at the aggregate level.

His model can be described as: In round t, player i has a propensity $q_{ij}(t)$ to choose pure strategy j, so that $p_{ij}(t) = \frac{q_{ij}(t)}{\Sigma_{k=1}^{N} q_{ik}(t)}$ is the probability that agent i plays strategy j in period t. The propensity q_{ij} is updated based on the payoff x received for choosing strategy j in the previous period:

$$q_{ij}(t+1) = q_{ij}(t) + (x - x_{\min}),$$

where $x_{\min}$ is the lowest possible payoff. Thus the propensity to choose a strategy is *reinforced* if it has provided higher payoffs in the past, and vice versa.

Roth & Erev (1995) and Erev & Roth (1998) generalized Arthur's RL model to get a better fit with experimental data from multi-player games. Their initial propensities $q_{ij}(1)$ are equal across all N strategies: $\Sigma_j q_{ij}(1) = S_i(1) = S(1)$, an initial propensity parameter, equal across all players and strategies. The rate of learning is proportional to $S(1)$. Again, $p_{ij}(t) = \frac{q_{ij}(t)}{\Sigma_{k=1}^{N} q_{ik}(t)}$ is the probability that agent i plays strategy j in period t.

Player i updates his propensity to play strategy j according to the rule:

$$q_{ij}(t+1) = (1-\phi)q_{ij}(t) + E_k(j, R(x)),$$

where $E_k(j, R(x)) = (1-\phi)R(x)$ if $j = k$, or

$= \frac{\varepsilon}{N-1} R(x)$ otherwise;

and where $R(x) = x - x_{\min}$.

There are thus three parameters in the Roth-Erev model: the initial-propensity parameter $S(1)$; the recency parameter ϕ, which reduces the power of past experiences; and the experimentation parameter ε. Arthur's RL model is a special case of the Roth-Erev model: when $\phi = \varepsilon = 0$, Roth-Erev is Arthur.

6. From Analysis to Design

Roth (1991) was the first to argue for the economist as designer: *the market engineer*. He outlined the iterative process of market design using three possible approaches:

1. traditional closed-form game-theoretic analysis;
2. experimental results from economics laboratories; and
3. computational exploration of different designs.

If the design criteria are clearly defined, some recent techniques of simulation and optimization from computer scientists and computational economists can be used to search for optimal market designs, although, following our discussion above, only in special cases can this be done directly.

Historical market institutions have in general not been imposed from above (*top-down design*) but have emerged from the bottom up as a consequence of a multitude of actions and interactions of the myriad traders (McMillan 2002).

The omnipotent programmer can experiment with different market forms and different kinds of boundedly rational agents to discover sufficient combinations of each for specific behavior of the market, But evolutionary computation raises the possibility of *bottom-up design*, or emergence of market design through simulation.

There may be difficulties in the design process, as we alluded to in the discussion of complexity above. Design is a process of building that is directed by the pre-specified design objectives, or the formal specifications. But specifying objectives does not at all resolve the issue of exactly how the model building should occur. Why?

Objectives are specified in a performance space (or behavior space), but the building occurs in a design space. The mapping from designed structure to the desired performance is not clear. In biological evolution, design occurs in the genome space, while behavior or performance occurs in the phenome space. In designer markets, policy-makers use theory, human experiments and computer simulations to analyze how successful the design process has been: the mapping from design (structure and rules) to behavior of the economic actors (the performance of the system). Where the mapping is sufficiently well understood, and where closed-form analytic solution is tractable, it is possible to describe not only *sufficiency* but also *necessity*.

Sufficiency: If the market has this structure, and the rules of trading are such and such and the traders are given this information, then this market performance and trader behavior will follow. Necessity: If you want this performance and behavior, then this is the only set (or sets) of designs (combinations of structure and rules) that will produce it.

With no closed-form analytical solution, but with human experiments or with computer simulations, necessity is in general unattainable, only sufficiency. But with few degrees of freedom, necessity is close: using copper rods and wooden "atoms," Watson & Crick (1953)[4] simulated the structure of DNA, given its chemical properties (acid), known atomic composition (and electrical properties), and with some X-ray diffraction photographs.

7. A Framework for Market Design

MacKie-Mason & Wellman (2006), who examine designing on-line, computerized markets, have given much thought to the elements of a market and the transactions undertaken in the market, in a lesson for traditional economists, who have sometimes taken some of these elements for granted, in examining traditional markets. Their work provides a framework for market design.

MacKie-Mason & Wellman characterise a transaction as including three fundamental steps: first, the connection (searching for and discovering the opportunity to engage in a market interaction), second, the deal (negotiating and agreeing to terms), and, third, the exchange (executing a transaction). They define a "marketplace system" as consisting of: the agents and the market mechanism through which they interact, all embedded in an environment of social institutions (language, laws, etc.).

Their market mechanism is the set of "rules, practices, and social structures of a social choice process, specifying: permissible actions" (including messages), and market-based exchange transactions as outcomes of a function of agent messages. (They are concerned specifically with ACE simulation.)

This characterization of a marketplace implies at least two design decisions: first, the design of the market mechanism, which might be decomposed into the design of mechanisms for, successively, the connection, the deal, and the exchange phases of a transaction, and, second, the design of agents to interact with the market mechanism, whether existing or newly designed.

---

4. Notice that the title of Watson & Crick's classic 1953 paper includes the phrase "*a* structure", not *the* structure: necessity had not been proved by the simulation, or "stereo-chemical experiment," in their words.

MacKie-Mason & Wellman (2006) define an agent as an "autonomous decision-making locus in a system of multiple decision-making entities". An agent has "type" attributes, such as preferences, beliefs, intentions, and capabilities. The designer wants consistency between the agents' behavior, beliefs, and preferences, consistent with some principle of rationality. In this paper we focus on the design of MacKie-Wellman's market mechanism, specifically the deal negotiation task that governs the settlement from allowable actions.

Mechanisms specify the agents' concerns that are recognized, and, the rules mapping actions into allocation outcomes. A rule might specify which actions are permissible, or the procedure for choosing a settlement of agents' concerns based on observable actions. For example, auctions have rules governing allowable actions, and rules governing settlement.

Designs are constrained in various ways. This means, in general, that the design of the market mechanism must be measured, and usually consists of a constrained optimization, even if not explicitly or directly. Examples of constraints on a market include: "No external subsidies" or "maintain horizontal equity".

The general market design issue has become designing a market mechanism that includes defining a set of concerns over which agents can interact, while specifying rules of permissible actions, and specifying rules for mapping from actions to settlement and outcomes.

8. Market Design

Design objectives are specified in a performance space (or behavior space) and the building occurs in a design space. The mapping from the designed structure to the desired performance may not be clearcut (because of the syntactic complexity discussed above) or even computable (as discussed above). For these reasons, direct design of markets (the programming problem as discussed above) is hardly ever attempted. Instead, we use iterative simulation (of the checking problem, as above) to determine a design or designs that result in the desired performance, subject to any design constraints.

Where there are several design criteria, the possibility arises of trade-offs between the criteria. For example, if a firm has market power, it can maximize its seller revenue, but at the cost of market efficiency, as measured by the sum of seller (or producer) surplus and buyer (or consumer) surplus. Or it might be possible to improve the fairness of a market outcome, but at the cost of market efficiency. Such trade-offs must be explicit.

It might be possible to use a version of Simon's (1981) satisficing: so long as the other criteria are met (above some target level), the remaining criterion is used to rank the designs. Or different criteria could be weighted to derive a single, scalar maximand.

How good is a designed auction market? Phelps et al., (2002, 2005) suggest eight possible criteria for comparing market designs: first, maximizing seller revenue: this was one of the main criteria in the design of the spectrum auctions, such as the 3G auctions (Milgrom, 2004); second, maximizing market allocative efficiency: a socially desirable policy attribute of a marketplace system; third, discouraging collusion, in order to attain the first and second criteria; fourth, discouraging predatory behavior, in order to help to maximize efficiency; fifth, discouraging entry-deterring behavior, in order to maximize seller revenue; sixth, budget balance: no third-party payments for a deal to be reached; seventh, individual rationality: the expected net benefit to each participant from the market mechanism should be no less than the best alternative; and, eighth, strategy-proofness: participants should not be able to gain from non-truth-telling behavior.

Analytical methods should be used where useful. For instance, Myerson & Satterthwaite (1983) derived an impossibility result: No double-sided auction mechanism with discriminatory pricing[5] can be simultaneously efficient, budget-balanced, and individually rational. LeBaron (2006), in examining the use of agent-based models of financial markets, discusses seven basic design questions for his models, which translate across to more general models. First, the economic environment itself needs to be resolved: what will be traded? what is the scope of the market? Second, how are agents' preferences to be modeled: with particular functional forms such as mean–variance, Constant Absolute Risk Aversion, myopic or inter-temporal, or perhaps just using evaluation of specific behavioral rules? Third, modeling of market clearing and price formation. Fourth, evaluating the fitness of the model: wealth or utility? And whether the evolving rules are forecast-based (what will the price be at time t?) or demand- and action-based. Fifth, how information is precessed and revealed. Sixth, how learning occurs: is it social and direct or at arm's length; is it individual? Seventh, how is benchmarking to be undertaken?

While these questions relate to the models used to design markets, they may also reflect on the design criteria for the final designer markets.

9. Agent-based Market Design

It is possible to design a market without the use of agents: for instance, given a market with demand and supply schedules, economic efficiency is maximized at the output level where marginal value equals the marginal unit cost, no matter how the social surplus is divided between buyers and sellers. But, as discussed above, such direct design (optimization) requires a well defined problem.

With several design trade-offs and the possible emergence of unforeseen performance in the system, enter agent-based analysis and design. ACE analysis and design models the market system as "evolving systems of autonomous, interacting agents with learning capabilities" (Koesrindartoto & Tesfatsion, 2005)

LeBaron (2006) places some weight on how actual trading occurs: the institutions under which trading is executed. He argues that agent-based models are well suited to examining market design and micro-structure questions for two reasons: first, they can produce a large amount of data, and, second, they allow testing of market design in a heterogeneous, adaptive environment.

Examples of analysis using ACE models include Audet et al. (2002), an agent-

5. In discriminatory-price auctions (or "pay-as-bid" auctions), distinct trades in the same auction round occur at distinct prices; in uniform-price auctions, however, all trades in any given auction round occur at the same price.

based study of stock-market micro-structure (order books v. dealers). Bottazzi et al. (2005) is another stock-market study that examines tick sizes (and unexpectedly determines that smaller tick sizes do not necessarily improve the market's efficiency). Chan & Shelton (2001) examine how a model behaves with different RL mechanisms, all of which enable the optimum policy function for a market-making broker to be found. Marks (1989*b*), Arifovic (1994), Midgley et al. (1997) modeled market interactions using a GA, as did Nicolaisen et al. (2000) who examined an electricity market. On the other hand, Nicolaisen et al. (2001) made a single change from GA learning to RL learning, for a better result, as Marks (2006) discusses at length.

A rare example of a case of direct design of a market (a single auction) is provided by Byde (2006). This is a sealed-bid auction where the highest bidder wins and pays an amount given by

$$(1 - w)bid_1 + wbid_2,$$

where bid_1 is the highest bid and bid_2 the second-highest. It is readily seen that when $w = 0$, it's a first-price auction, and when $w = 1$, a second-price auction. Using a GA to explore the impacts on seller's revenue, Byde found under certain plausible conditions that seller's revenue is maximized when $w = 0.3$. Thus he derived a new, "synthetic" auction, superior (for the seller) to both first-price and second-price auctions.

10. Barriers to ACE Acceptance and Simulation

Leombruni & Richiardi (2005) question the evident reluctance of main-stream economists to embrace ACE modeling. They found only 8 ACE articles among the 26,698 in the top 20 economics journals[6] from 1970 to 2004.

They give two possible reasons: first, difficulties in the interpretation of the simulation dynamics and in generalization of the results, and, second, problems in estimation of the simulation models. I would add: in general, *no necessary conditions* from simulation, just sufficient conditions, and the need for validation of the models.[7]

In the 2006 *Handbook* (Tesfatsion & Judd 2006), a search reveals that only 4 of 24 chapters mentioned "validation", a total of 9 times. This unfortunate statistic does not reflect the importance of validation in ACE.[8] There are two related activities in properly and effectively using models, whether simulation models in general, ACE models in

---

6. These include, in order (Kalaitzidakis et al. 2003): *American Economic Review, Econometrica, Journal of Political Economy, Journal of Economic Theory, Quarterly Journal of Economics, Journal of Econometrics, Econometric Theory, Review of Economic Studies, Journal of Business and Economic Statistics, Journal of Monetary Economics, Games and Economic Behavior, Journal of Economic Perspectives, Review of Economics and Statistics, European Economic Review, International Economic Review, Economic Theory, Journal of Human Resources, Economic Journal, Journal of Public Economics, Journal of Economic Literature.* I note that at least four articles on ACE models have appeared in these journals so far in 2005, a good sign.

7. I am grateful to Nick Vriend for pointing out that very few articles presenting traditional closed-form models have included validation.

particular, or, indeed, closed-form models: verification and validation.

Verification (sometimes known as internal validity) asks: is the simulation working as the modeler wants it to? — is it "doing the thing right?" Validation asks: is the model used in the simulation the correct model? — is it "doing the right thing?" (Boehm 1981).

To verify: use a suite of tests, and run them every time you change the simulation code — to verify the changes have not introduced extra bugs. To validate: ideally, compare the simulation output with the real world. But there are two reasons why this comparison might fail. First, the two processes, real-world and model, are *stochastic,* which means that complete accord is unlikely, and the distribution of differences between the two processes is usually unknown. Second, the two processes are *path-dependent,* which means that output from either process is sensitive to initial conditions/parameters. Another means of assuring oneself that the model is correct — of validating the model — is the test the model for "retrodiction," which means reversing time in the simulation, starting from a known date and attempting to match earlier behavior between the historical real world and the model's output. Finally, what if the model is correct, but the input data are bad? Then the model will faithfully reflect its given inputs, but in that case the output will not accord with reality, at any level (see below). Further, the modeler can use sensitivity analysis, to ask: first, about the robustness of the model to assumptions made, and, second, which the crucial initial conditions/parameters are. This sensitivity can include randomized Monte Carlo stochastic sampling, with many runs.

Judd (2006) quotes statistician John Tukey, who, in 1962, said, "Far better an approximate answer to the right question ... than an exact answer to the wrong question." That is, economists face a tradeoff between: the numerical errors of computational work and the specification errors of analytically tractable models. Judd makes some further suggestions: First, the simulator should search for counterexamples: if found, then they should provide insights into when the proposition fails to hold. If not found, then the simulator does not have a proof (remember: sufficiency, but not necessity), but failure to find counterexamples should provide strong evidence for the truth of the proposition. Second, the simulator should use sampling methods: Monte Carlo, and quasi-Monte Carlo methods (Judd 1998), which will allow the use of standard statistical tools to describe the confidence in the results. Third, use of regression methods to find the "shape" of the proposition. Fourth, replication of the model and generalization of the results: this has led to so-called"docking" (Axelrod 2003) by replicating on a different platform or language, although lack of standard software is an issue. Finally, Judd argues that there are productive synergies between simulation and conventional theory, which should be exploited. The Myerson & Satterthwaite impossibility result (above) which reduces the space of possible feasible specifications is an example of such synergy.

Axelrod (2003) discusses model replication and "docking" of simulation models. By *docking* he means that a simulation model written for one purpose is aligned or "docked" with a general purpose simulation system written for a different purpose. From his experiences with model docking, he draws four lessons: First, model docking is not necessarily so hard. Second, replication can be measured at three decreasingly exact

8. One of the *Handbook* editors, Leigh Tesfatsion, informs me that there had been plans for a chapter on validation, sadly unfulfilled.

levels: (*a*) numerical identity, (*b*) distributional equivalence, and (*c*) relational equivalence. Third, which is the null hypothesis? And what is the sample size? Fourth, minor procedural differences (e.g. sampling with or without replacement) can block replication, even at the level of distributional equivalence.[9]

Axelrod gives four reasons for errors in docking occurring: First, there was ambiguity in the published model descriptions. Second, there were gaps in the published model descriptions, which were incomplete. Third, there were errors in the published model descriptions. Fourth, there were (as mentioned above) software and/or hardware subtleties, such as. different floating-point number representations in the two computers.

Validation is difficult, especially with a model of a complex system (which can result in emergence) (Kelton et al. 2001); and when the parameter space is large (Shervais et al. 2003); with possible path dependence, positive feedback, extreme sensitivity to initial conditions; and with incomplete knowledge of micro-details.

LeBaron (2006) suggests three steps to validation. First, attempt to replicate difficult empirical features: do ACE models fit facts not otherwise explained? Two, put parameters under evolutionary control: such things as learning rates and memory depth (when using an evolutionary model). Third, use results from laboratory experimental markets with human subjects: this can elucidate the learning dynamics (such as refinements to RL models) for ACE models.

I would argue in conclusion that ACE modelers must try harder: the challenge of validation to gain acceptance is an opportunity to demonstrate the relative indifference of the closed-form traditionalists to validation. There is a need for: benchmarking: both against history, and against other models (with docking); seeking the extremes or "breaking" the model (Miller 1998): what levels of inputs (separately or in combination) result in absurd outputs? looking at the model as a "black box" and exploring its response to step functions (off and on, minimum and maximum, one input variable at a time), and statistically estimating the model as a function from inputs to outputs (inputs as independent variables, outputs as dependent variables).

Judgment of the modeler should result in acceptance by the policy-makers, but how? The modeler should first convince herself of the model's validity, as the most skeptical observer.

11. Conclusions

Market Design in the face of complexity in the mapping from initial conditions (structure, parameters) in the design space to behavior in the performance space requires iterated analysis to explore the mapping. Simulations allow the modeler to tackle the programming problem of determing to what extent given market specifications (structure, rules and environment) will result in the desired market performance and behavior. Used

---

9. Marks (1989*a*) was a replication or docking of Axelrod (1987); whereas Axelrod wrote in Pascal VS, Marks wrote in C on different hardware; the replication was at least relationally equivalent, if not numerically identical. Marks & Schnabl (1999) was a replication of Axelrod (1987) using a GA and a Neural Net.

iteratively, this process allows the modeler to explore the space of feasible market designs. and to choose a desired design, before the real-world market is undertaken. Using bottom-up ACE modeling can capture heterogeneity, as well as modeling how behavior co-evolves in competitive interactions.

The use of simulations in general, and ACE simulations in particular, to design markets requires validation of the models used. The paper discusses heuristics for validation, which is ultimately an issue of confidence in the model and the modeler.

In his Fisher-Schultz Lecture, Roth (2002) argued that "in the service of design, experimental and computational economics are natural complements to game theory," and that the economics literature should encompass market design experience in sufficient detail to allow scientific knowledge about market design to accumulate in the discipline. Indeed, he remarks that although the existing theoretical literature gave only broad directions in his task of designing the labor clearinghouse for American doctors, his use of experimental results and computational models made this design possible, and incidentally showed that answers from static theory were often close to the experience of a dynamic market. It is in the spirit of Roth's call to action that this paper has been written.

Acknowledgements.

I wish to ackowledge advice and comments from Leigh Tesfatsion, Peter McBurney, David Midgley, Shu-Heng Chen, and participants at seminars at the University of Michigan, the University of Bielefeld, the University of Birmingham, the National Chengchi University of Taipei, and the National Kaohsiung University of Applied Sciences. A first draft of the paper was written at INSEAD, Fontainebleau. Some of the ideas were reinforced when I gave the Fourth Herbert Simon lecture series in Taiwan.

References

[1] Arifovic, J. (1994), Genetic algorithm learning and the cobweb model, *Journal of Economic Dynamics and Control,* 18: 3–28.

[2] Arthur, W.B. (1991), Designing economic agents that act like human agents: a behavioral approach bounded rationality, *American Economic Review,* 81: 353–359.

[3] Arthur, W.B. (1993), On designing economic agents that behave like human agents, *Journal of Evolutionary Economics,* 3: 1–22.

[4] Arthur, W. B., J. Holland, B. LeBaron, R. Palmer, and P. Tayler (1997), Asset pricing under endogenous expectations in an artificial stock market, in W. B. Arthur, S. Durlauf and D. Lane, (eds.), *The Economy as an Evolving Complex System II*, Reading, MA: Addison-Wesley, pp. 15–44.

[5] Axelrod R. (1987), The evolution of strategies in the iterated Prisoner's Dilemma, in L. Davis (ed.), *Genetic Algorithms and Simulated Annealing,* London, Pittman.

[6] Axelrod, R. (2003), Advancing the art of simulation in the social sciences, *Japanese Journal for Management Information Systems,* 12(3), Dec.

[7] Audet, N., Gravelle, T., and Yang, J. (2002), Alternative trading systems: does one shoe fit all? Bank of Canada, Working Paper 2002-33.

[8] Axtell, R. (2005), The complexity of exchange, *The Economic Journal*, 115 (June), F193–F210.

[9] Boehm, B.W. (1981), *Software Engineering Economics*, Prentice-Hall.

[10] Bottazzi, G., G. Dosi, and I. Rebesco (2005), Institutional architectures and behavioral ecologies in the dynamics of financial markets, *Journal of Mathematical Economics,* 41 (1–2): 197–228.

[11] Bunn, D.W., and F.S. Oliveira (2003), Evaluating individual market power in electricity markets via agent-based simulation, *Annals of Operations Research,* 121: 57–77.

[12] Byde, A., (2006), Applying evolutionary search to a parametric family of auction mechanisms, *Australian Journal of Management*, 31(1), June, forthcoming.

[13] Chan, N.T., and C. Shelton (2001), An electronic market-maker, Artificial Intelligence Lab, MIT, AI Memo 2001-005, April.

[14] Chattoe, E. (1998), Just how (un)realistic are evolutionary algorithms as representations of social processes? *Journal of Artificial Societies and Social Simulation,* 1(3). <http://www.soc.surrey.ac.uk/JASSS/1/3/2.html>

[15] Curzon Price, T. (1997) Using co-evolutionary programming to simulate strategic behavior in markets, *Journal of Evolutionary Economics,* 7(3): 219–254.

[16] Dawid, H. (1999), *Adaptive Learning By Genetic Algorithms: Analytical Results and Applications to Economic Models,* Berlin: Springer, 2nd ed.

[17] Duffy, J., (2006), Agent-based models and human-subject experiments, In Tesfatsion & Judd, op.cit.

[18] Edmonds, B. & Bryson, J. (2004) The insufficiency of formal design methods — the necessity of an experimental approach for the understanding and control of complex MAS. In Jennings, N. R. et al. (eds.) *Proceedings of the 3rd International Joint Conference on Autonomous Agents & Multi Agent Systems* (AAMAS'04), July 19-23, 2004, New York. ACM Press, pp. 938-945.

[19] Erev, I., and A.E. Roth (1998), Predicting how people play games: reinforcement learning in experimental games with unique mixed strategy equilibria, *American Economic Review,* 88(4): 848–881.

[20] Gödel K. (1931), Über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme I, *Monatshefte für Mathematik und Physik*, 38: 173–198.

[21] Hailu, A., and S. Schilizzi, (2004), Are auctions more efficient than fixed price schemes when bidders learn? *Australian Journal of Management*, 29(2): 147–168, December.

[22] Janssen, M.A., and E. Ostrom (2006), Governing socio-ecological systems, in Tesfatsion & Judd, op. cit.

[23] Jennings, N.R., P. Faratin, A.R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge (2001), Automated negotiation: prospects, methods, and challenges, *International Journal of Group Decision and Negotiation,* 10(2): 199–215.

[24] Judd, K.L. (1998), *Numerical Methods in Economics,* Camb., MIT Press.

[25] Judd, K.L. (2006), Computationally intensive analyses in economics, in Tesfatsion & Judd, op. cit.

[26] Kalaitzidakis, P., T. Stengos, T.P. Mamuneas (2003), Rankings of academic journals and institutions in economics, *Journal of the European Economics Association,* 1(6): 1346–1366.

[27] Kauffman, S.A., (1995), *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity,* NY: OUP.

[28] Kelton, W.D., (2001), Some modest proposals for simulation software: design and analysis of experiments, Dept. of Quantitative Analysis & Operations Management, University of Cincinnati.

[29] Koesrindartoto, D., and L. Tesfatsion (2004), Testing the reliability of FERC's Wholesale Power Market Platform: an agent-based computational economics approach, Staff General Research Papers No. 12326, Iowa State University, Department of Economics.

[30] Kunchamwar, D., Marks, R., & Midgley, D. (2005) Destructive verification & empirical validation of an agent-based simulation of consumers, retailers & manufacturers, UCLA, Lake Arrowhead, May.

[31] LeBaron, B., (2006), Agent-based computational finance, in Tesfatsion & Judd, op. cit.

[32] Leombruni, R., and M. Richiardi (2005), Why are economists sceptical about agent-based simulations? *Physica A,* 355: 103–109.

[33] MacKie-Mason, J.K., and M.P. Wellman (2006), Automated markets and trading agents, in Tesfatsion & Judd, op. cit.

[34] Marks, R.E. (1989*a*), Niche strategies: the Prisoner's Dilemma computer tournaments revisited, mimeo. AGSM Working Paper 89-009.

[35] Marks, R.E. (1989*b*) Breeding optimal strategies: optimal behavior for oligopolists, *Proceedings of the Third International Conference on Genetic Algorithms,* George Mason University, June 4–7, 1989, ed. by J.D. Schaffer, San Mateo, Calif.: Morgan Kaufmann Publishers, pp.198–207. Also published in the *Journal of Evolutionary Economics,* 2: 17–38, 1992.

[36] Marks, R.E., (1998), Evolved perception and behaviour in oligopolies, *Journal of Economic Dynamics and Control,* 22(8–9): 1209–1233.

[37] Marks R.E., (2006), Market design using agent-based models, Ch 12 in Tesfatsion & Judd, op. cit.

[38] Marks R.E., and Schnabl H. (1999), Genetic Algorithms and Neural Networks: a comparison based on the Repeated Prisoner's Dilemma, Thomas Brenner (ed.), *Computational Techniques for Modelling Learning in Economics,* Dordrecht: Kluwer Academic Publishers, pp. 197–219.

[39] McMillan, J. (2002), *Reinventing the Bazaar: A Natural History of Markets,* NY: Norton.

[40] Midgley, D.F., R.E. Marks, and L.G. Cooper (1997), Breeding competitive strategies, *Management Science*, 43(3): 257–275.

[41] Milgrom P. (2004), *Putting Auction Theory to Work,* Cambridge: Cambridge University Press.

[42] Miller, J.H. (1998), Active nonlinear tests (ANTs) of complex simulation models, *Management Science*, 44(6): 820–830.

[43] Myerson, R.B., and M.A. Satterthwaite (1983), Efficient mechanisms for bilateral trading, *Journal of Economic Theory,* 29: 265–281.

[44] Nicolaisen, J., M. Smith, V. Petrov, and L. Tesfatsion (2000), Concentration and capacity effects on electricity market power, *Proceedings of the 2000 Congress on Evolutionary Computation,* ed. by A. Alzala, Piscataway, N.J., IEEE Society Press, pp. 1041–1047.

[45] Nicolaisen, J., V. Petrov, and L. Tesfatsion (2001), Market power and efficiency in a computational electricity market with discriminatory double-Auction pricing, *IEEE Transactions on Evolutionary Computation,* 5(5): 504–523.

[46] Phelps, S., P. McBurney, S. Parsons, and E. Sklar (2002), Co-evolutionary auction mechanism design: a preliminary report, *Lecture Notes In Computer Science: Revised Papers from the Workshop on Agent-Mediated Electronic Commerce IV, Designing Mechanisms and Systems,* ed. by J. A. Padget, O. Shehory, D. C. Parkes, N. M. Sadeh, and W. E. Walsh, Berlin: Springer, pp. 123–142.

[47] Phelps, S., S. Parsons and P. McBurney (2005), Automated trading agents versus virtual humans: an evolutionary game-theoretic comparison of two double-auction market designs, in: P. Faratin and J.A. Rodriguez-Aguilar, eds., *Agent-Mediated Electronic Commerce VI: Theories for and Engineering of Distributed Mechanisms and Systems,* Lecture Notes in Computer Science, (Springer, Berlin).

[48] Roth, A.E., (1991), Game theory as a part of empirical economics, *Economic Journal,* 101(401): 107–14.

[49] Roth, A.E. (2002), The economist as engineer: game theory, experimentation, and computation as tools for design economics, *Econometrica,* 70(4): 1341–1378.

[50] Roth, A.E., and I. Erev (1998), Predicting how people play games: reinforcement learning in experimental games with unique mixed strategy equilibria, *American Economic Review,* 88(4): 848−881.

[51] Shervais, S., Wakeland, W., Raffo, D., (2003), Evolutionary verification and validation of software process simulation models, mimeo. <http://prosim.pdx.edu/prosim2004/abstract/wakeland_ext_abs.pdf>

[52] Simon, H.A., (1981), *The Sciences of the Artificial,* 2nd ed., Cambridge: The MIT Press.

[53] Simon, H.A. (1983), Why should machines learn?, in *Machine Learning: An Artificial Intelligence Approach,* R.S. Michalski, J.G. Carbonell and T.M. Mitchell, (eds.), Springer-Verlag. Berlin. pp. 25−37.

[54] Tesfatsion, L., and Judd, K.L., (eds.) (2005), *Handbook of Computational Economics, Volume 2: Agent-Based Computational Economics*, North-Holland, forthcoming.

[55] Turing A.M. (1936) On computable numbers, with an application to the *Entscheidungsproblem*, *Proceedings of the London Mathematical Society,* 42: 230−65, 43: 554−6.

[56] Vriend, N., (2000), An illustration of the essential difference between individual and social learning and its consequences for computational analyses, *Journal of Economic Dynamics and Control,* 24: 1−19.

[57] Watson, J.D., and F.H.C. Crick (1953), Molecular structure of nucleic acids: A structure of deoxyribose nucleic acid, *Nature,* 4356: 737−8, April 25.