Emergent Economies for Role Playing Games

Jonathon Doran, Ian Parberry
Dept. of Computer Science & Engineering
University of North Texas

***Abstract*—We present a novel application of economics to the problem of determining prices for goods and services within computer role playing games, particularly those with persistent worlds. These games typically determine prices *a priori*, and leave them fixed for the game's duration. Unlike the real world, games are judged by the amount of interest they generate within players, and we believe that allowing prices to be changed in response to player actions adds interesting gameplay elements. The field of economics provides a rich body of literature discussing how prices may be determined, and we have adapted these ideas into an economic model suitable for use with role playing games. Our price update algorithms provide interesting behaviour, while preserving stability under a range of conditions. Our experiments simulate daily trading over 5 year intervals in order to produce evidence of short-term and long-term behavior. Our results indicate that the novel economic model we present is sufficiently stable, resilient, and consistent with behaviors of real world economies to merit implementation in role playing games.**
Keywords: Economics, Autonomous Trading Agents, Role-Playing Games, Machine Learning, Agent Intelligence.

I. Introduction

While role playing games such as Oblivion, Fallout, Everquest, and World of Warcraft allow players to trade goods and services with computer controlled non-player characters (NPCs), acceptable prices in these games have been selected *a priori* by game designers and typically remain fixed throughout the game. Game economies differ from real world economies in that interesting behavior is more desirable than accurate modeling, and static prices do not seem as interesting as those that change as the result of game events. Little work has been done to apply traditional economics to role playing games, therefore we present a price update system, grounded in traditional economics, that may be used in such games. In particular we describe the algorithms and the parameters in sufficient detail for the reader to duplicate our work.

One of the drawbacks of such a static economy is that nothing changes, and in particular the player is unable to cause changes to occur. Under a static economy, vendor preferences do not change and as a result one sees situations where vendors have a seemingly infinite supply and demand for commodities. In a dynamic economy trades fulfill needs, leading to the adoption of new preferences, and new behaviors. In particular we expect traders to buy only those commodities that are useful to them, and if they are unable to trade profitably to go out of business and find another line of work.

A game economy consists of a set of players and NPC agents that periodically trade with each other. These agents take on the role of vendors, as any of these agents could at a particular point in time offer to buy or sell a commodity or service from any other trader. This set will be referred to as a *market*, and the participants will be known as *traders*. Offers to buy a good or service will be referred to as *bids*, and offers to sell a good or service will be known as *asks*.

While the most general form of an economy involves the trade of both goods and services, we choose to simplify this model without loss of generality by considering all trades to involve only goods, referred to as *commodities*. We are able to do this because a service can be mapped to a tradable token such that the token may later be exchanged for the performance of the service. In the real world we see this same type of conversion used with postage stamps and gift cards for pure services such as haircuts.

In addition to the determination of prices, supply, and demand, an economic system will determine the allocation of resources. We consider both commodities and the NPC traders themselves to be resources, the latter because the market is able to generate wealth through them. Each NPC trader may be allocated or assigned one of several roles that govern its behavior, and we will see later that if the market is in equilibrium the allocation is Pareto-optimal (meaning that no trader may improve its position by modifying an action without making another trader worse off, see Pareto [1]). This is a useful observation since it gives us the ability to determine a reasonable distribution of roles within a community. We are thus able to determine the exact number of agents for each role that may be supported given current market conditions, and also to determine when an agent is no longer contributing to the economy.

Given an allocation of NPCs to various roles within the game, we indirectly determine what commodities will be purchased or sold. Supply and demand determine what roles are profitable, and the allocation of NPCs to these roles determines future supply and demand. Thus the simulated economy is a feedback loop, ideally with the behavior of all traders being

The authors are with the Dept. of Computer Science & Engineering, Univ. of North Texas, 1155 Union Circle #311366, Denton, Texas 76203–5017, U.S.A. (email: `jhd@unt.edu`, ~~`ian@unt.edu`)~~. `http://ianparberry.com`

interrelated.

In evaluating our experimental results, we consider long-standing works in economics by Adam Smith [2] and Vernon Smith [3], [4] and find the behaviors of our model to be sufficiently consistent with classic economic theory to confirm the usefulness of our approach. We have also studied the price update behavior of over 1 billion simulated economies, and are satisfied that the variability and consistency of our results is sufficiently interesting to appeal to game players.

We will demonstrate an economic system with the following properties:

- It determines internally consistent prices for a variety of commodities.
- It adapts to external perturbations and shocks.
- It determines allocations of agents to roles.
- It is not dependent on any one set of game rules.
- It requires little memory or computation per transaction.

The goal of a game economy is to provide the player with an interesting experience that changes over time, and one in which they feel they have some measure of influence. Our work differs from traditional computational economics by stressing interesting behavior at the expense of realism. As with advances in areas such as computer graphics, we note that players are willing to suspend their disbelief if they enjoy the game experience. We show how to implement a system that can provide interesting reactive behavior, is easily added to games, and has low resource requirements. This type of economic behavior is lacking in most current role playing games, and it is our belief that future games would benefit from a richer set of economic behaviors.

The remainder of this paper is divided into nine sections. In section II we discuss background work on related topics. In section III we present our economic system. In section IV we discuss our experimental procedure, including details on the simulator used to evaluate the system. In section V we discuss agent replacement and the allocation of agents to roles. In section VI we describe a representative ruleset, and in section VII we discuss the use of randomly generated rulesets. Section VIII presents the results of these experiments along with an analysis. Finally in section IX we summarize our conclusions and discuss areas suitable for further study.

II. Related Work

Researchers at Iowa State University have done a lot of work in the area of agent-based computational economics, and this has drawn our attention, particularly the price resolution technique found in Nicolaisen, Petrov, and Tesfatsion [5]. Their work tends to focus more in qualitative areas, such as learning relationships between factors in a market, while we are concerned primarily with quantitative results: what is a good price for a commodity at this point in time? Steiglitz [6] has also performed agent-based simulations, however based on inspection of the source code we feel that our agents behave more rationally (for example, their agents appear to purchase as much of a commodity as they can afford without regard to the price or the agent's need).

The TAC-SCM (Trading Agent Competition - Supply Chain Management) competition has been ran annually since 2003, and produced many papers in the area of supply-chain management (see, for example, [7], [8], [9], [10]). In this competition, agents buy and sell commodities and produce products for resale. They attempt to predict changes in prices, and operate in a profitable manner. While supply-chain management is a substantially different problem, some of the techniques used by these simulations are of interest to us.

Roth and Erev [11] used reinforcement learning (RL) to learn prices in a simulated economy. In particular they used the acceptance or rejection of offers to provide reinforcement of a trading agent's pricing policy. We considered this approach, but were concerned that the amount of state we would need to consider would make policy convergence impractical. Flores and Garrido [12] similarly used RL, and we experimented with their technique of linearly interpolating prices using weights on the low and high end of the price range.

One advantage of reinforcement learning is the ability to update policies in an environment where agents do not know their current state. Value-based approaches like Q-learning or temporal difference (TD) learning do not work well in environments with hidden state, as agents need to know the current state in order to select a corresponding action. However, one is able to create an equivalence set of states based on observations of the system, and estimate which set likely contains the current state. Dahl's work with poker [13] shows that RL can work with hidden state, however for this problem the current visible state allows one to know the exact trajectory through action space that has been taken so far.

Price determination for a set of commodities is a significant problem, and a variety of techniques have been used with other problems. Several TAC-SCM entrants (for example, [7], [14], [15]) attempted to predict winning bids. The Botticelli trader estimated the probability of filling orders, and adjusted offer prices until the expected trade volume matched its ability to fill them. The probabilities are updated based on trading experience, which makes price a function of market history. Many more factors could go into pricing (utility, pricing trends, market supply and demand). There is also the question of whether probability is a linear function. Within a small range, linear approximations are adequate, but with larger uncertainties a nonlinear update may be more appropriate.

Pardoe and Stone [14] used a Bayes classifier to estimate the probability of an offer being accepted, and trained their classifier using data from prior TAC scenarios. In our case, the state of other agents is not known, limiting our ability to estimate prices. We chose instead to base offers on each agent's belief in the true price of a commodity, and not consider whether other agents might agree. We chose this approach to more closely model the imperfect information real traders have, and in turn we hope this leads to more realistic results.

Shapire [15] modeled price changes as a conditional density estimation problem. A price range was discretized into a set of bins, and a probability distribution was created over this set. This technique also modeled future prices as a function of historical prices, which works well if there are no other factors that might affect prices.

Wellman *et al.* [16] used a novel approach by estimating future demand for a commodity and adjusting prices in advance of the market.

Ketter [17] inferred market regimes (conditions such as oversupply of a commodity) based on the results of attempted trades. Gaussian Mixture Models (GMMs) were fitted to historical price data, and used as classifiers. We did not require these types of predictions, since our problem was defined as allowing agents access to market statistics (for resolved offers), or market price history. Ketter's system modeled price as a function of demand (similar to Wellman) and estimated trade volume as being similar to previous rounds.

Studies have been made of the economies of various massively multiplayer online role-playing games (MMORPG) by Simpson [18] and Castronova [19]. While these do not tell us how to simulate game economies, we see that game economies do behave similarly to real-world economies.

Meadows [20] developed models and a simulation to study social systems, however one could argue that these were economic models since they addressed resource allocation, and growth. This is also an excellent overview of model creation and simulation, we note in particular how one must describe the type of information a model is intended to produce. Following Meadows' categorization, our model provides projections of dynamic behavior modes. We omit the term "imprecise" as the prices reported represent actual trades in the simulation, and would presumably be actual NPC offer prices when used in a game.

III. An Economic System

A simulated economic system serves four major functions as shown in Figure 1. In addition to determining the prices of commodities it also determines order quantities (supply and demand for each commodity), the production and consumption of commodities (indirectly related to supply and demand), and an allocation of commodities and roles to participating agents. This mirrors the properties of real economic systems, in particular the coupling of price with supply and demand as discussed by Adam Smith [2].

Each agent maintains a set of price beliefs for each commodity it is able to buy or sell. These price beliefs are represented as an upper and lower price bound, with the agent believing the price to be somewhere in this interval. Any time the agent needs to make a price estimation (for example during offer creation), it will select a uniformly random value in this interval. The outcome of a trade will provide either positive or negative reinforcement to this belief. Positive reinforcement will result in the agent shrinking this interval around the mean, negative reinforcement may result in the interval increasing about the mean and possibly being translated to a different mean. A designer interested in creating an economic system would need to decide when these updates occur, and the magnitude of the changes.

Periodically agents will need to submit trade offers to the clearing house in order to buy or sell commodities. When an agent wishes to create an offer, it will need to determine the commodity to trade, a fair price, and the quantity of the commodity to trade. A designer may choose to have agents buy only commodities they use for production and sell commodities they produce. In this case, an agent would create bids when the inventory of needed commodities drops below some threshold, and create asks anytime it has inventory to sell. The CREATE BID routine creates an offer to buy at most $limit$ units of $Commodity$, and CREATE ASK creates an offer to sell at least $limit$ units of $Commodity$.

CREATE BID$(Commodity, limit)$

bid-price $=$ PRICEOF $(Commodity)$
ideal $=$ DET-PURCHASE-QUANTITY$(Commodity)$
quantity-to-buy $=$ Min(*ideal*, $limit$)

CREATE ASK$(Commodity, limit)$

bid-price $=$ PRICEOF $(Commodity)$
ideal $=$ DET-SALE-QUANTITY$(Commodity)$
quantity-to-sell $=$ Max(*ideal*, $limit$)

The determination of offer quantities is based on an agent's need, the inventory on hand, and the observed market price for that commodity. An agent might determine that it has no need to trade in a particular commodity, or that a need is present but current market prices are unfavorable and trades should be avoided. If an agent believes that a commodity is either overpriced or underpriced, it will adjust the quantity in its order depending on whether the agent is buying or selling. The quantity is scaled based on the location of the current market price within the trading range that the agent has observed. Agents that trade more frequently will have observed more trades, and will therefore have a better idea of the trading range. Agents that trade infrequently are more likely to make mistakes in pricing, however the resolution of these trades will cause the price history to be updated and the agent will improve its performance in future trades.

While there are likely many different means of determining trade quantity, we have had success basing this number on how far the agent's price belief is

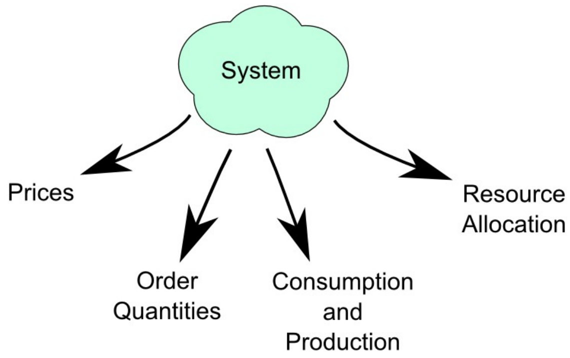

Fig. 1. Responsibilities of an economic system.

from the observed market average. This introduces an important but subtle distinction, as prices may be expressed in two different forms. A historical average price represents successful trades that have occurred in the past. Agents should be aware that past performance is no guarantee of future results, and should therefore trust their own beliefs more than the historical average. However, agents should question their belief if these values diverge and their offers are being rejected.

DET-SALE-QUANTITY($Commodity$)
$mean$ = historical mean price of $Commodity$
$favorability$ = position of *mean* within observed trading range
amount to sell = *favorability* * excess inventory of Commodity
return *amount to sell*

DET-PURCHASE-QUANTITY($Commodity$)
$mean$ = historical mean price of $Commodity$
$favorability$ = max price - position of *mean* within observed trading range
amount to sell = *favorability* * available inventory space
return *amount to sell*

An economic model may be used as a tool for allocating resources, determining trade volumes, or estimating commodity prices. These factors may be expressed as a set of coupled functions of the other factors. In general, supply and demand determine price, and price determines supply and demand (Smith [2]). For example, in an economy where wheat is sold, the amount of wheat traded on the market is a function of the bid and ask prices, and the quantity traders are willing to trade at these prices.

The amount of each commodity that is produced and consumed is determined by having each agent PERFORM-PRODUCTION. In the most general form the agent will attempt to transform one basket of commodities into a second basket of commodities. The commodities that are consumed represent the raw materials used up during production, and the commodities that are produced represent the products of the agent's labor. Production may therefore reduce an agent's inventory of raw materials, and increase its inventory of products. As a result of an inventory reduction, the agent may later find that its inventory is too low and create bids in an attempt to replace this inventory. This creates demand in the market, and the agent competes against other buyers for whatever supply is available on the market. Similarly, any increase in inventory may prompt the agent to create asks in an attempt to sell excess inventory. This creates supply in the market, and the agent competes against other sellers for whatever demand is available. There is therefore a strong relationship between the results of PERFORM-PRODUCTION and the supply and demand for various commodities in the market. This in turn implies a strong relationship with future prices for these commodities, as an increase in supply will tend to drive prices downwards and an increase in demand will tend to drive prices upwards.

It is important to note that there is no single correct price for a commodity, but rather a price that is acceptable to the community at a particular point in time. The individual trade prices may not be an optimum price, nor indicate a market equilibrium, as noted by Vernon Smith [3]. Traders will sometimes trade at non-optimal prices, and that they will learn from their experiences and adjust their future behavior, or they will fail and be removed from the market. While the actions of individuals are not optimal, they are usually rational and reflect the self-interest of each individual. These individuals tend to adjust their behaviors until they collectively behave in a Pareto-efficient manner.

An economic system also serves to allocate resources within the market. We have seen how agents compete to buy and sell commodities, and as a successful offer results in a trade it will also result in an allocation of the commodity traded to the buyer. A buyer who offers a higher price will have their offers accepted before lower priced offers, and therefore the market can be seen to allocate resources first to those who will pay more for them.

Each trading agent is assigned a particular role or profession when it is created, and maintains this role during its lifetime. This role determines the production rules that an agent will use when PERFORM-PRODUCTION is called, and subsequently the commodities that the agent will trade in. In the most general case, agents would not have these restrictions, but as we are also interested in determining a stable distribution of agent types, requiring agents to adhere to a limited set of rules allows us to make statements about the ability of a particular ruleset to support a given number of agents.

As the simulation progresses, successful agents will buy raw materials and sell their products. Unsuccessful agents will fail in their attempts to buy or sell, and therefore generate no cash flow. We have found that it is helpful to assess some fixed overhead, either in the form of required consumption, or in taxes, to pressure each agent to be productive. Under such a system unsuccessful agents will eventually go bankrupt as their money supply is exhausted, while successful agents will earn a profit above their expenses. When an agent goes bankrupt we choose to replace the agent with one of a profitable type, thereby adjusting the distribution of roles within the population of agents. This represents a market allocation of agents to roles.

A. Price Belief Updates

Agents will update their price beliefs in response to their offers being accepted or rejected, therefore agents that make offers gain more information on the market. These price beliefs are represented by a lower and upper bound for a price interval, with the agent believing the true price of a commodity lies within this interval. Agents are able to expand, contract and translate this interval as desired. An agent's price beliefs are updated after each of the agent's offers has been resolved. During this resolution, the system determines if a trade occurs, the number of units that traded, and the trade price. The offer resolution mechanism will be discussed in detail in the next section.

When an agent's offer is accepted, this is taken as evidence that the agent's price belief is accurate, and when an offer is rejected the agent learns that their price belief is inconsistent with the market. Even in the case of an accepted offer, it is beneficial for an agent to anticipate future price changes due to supply/demand imbalances. The price belief update performed will depend on whether the agent has generated a bid or an ask. The update procedures are described by PRICE-UPDATE-FROM-BID and PRICE-UPDATE-FROM-ASK. These updates take into account partially filled orders, the difference between the offer price and the historical market average, and the current supply and demand for the commodity being traded.

When an offer is accepted, the may only need to take supply and demand into account for a minor change. When an offer is rejected, the agent has a more difficult choice. The agent may have offered far from the mean, causing its offer to be placed far enough down in the offer book that no matching offer could be found, or the offer may have exceeded the limits of the matched agents. No seller will agree to sell a product below the cost to produce that product, nor will any agent agree to pay significantly above the observed trading range. In this case the rejected agent will want to reevaluate its price belief, translating its price range towards the mean and increasing the size of the interval to reflect its lack of confidence in the belief.

Agents that are very low on inventory and have had their bid rejected will make a more aggressive adjustment of their price belief in an attempt to leapfrog their competitors.

If none of these special situations exist, a rejected agent will examine the current round's supply and demand for the commodity and if there is a large imbalance adjust their prices in anticipation of price adjustments by potential trading partners.

```
PRICE-UPDATE-FROM-BID(Commodity)
if at least 50% of offer filled
    Move price belief limits inward by
    1/10 of upper limit
else
    Increase upper belief limit by
    1/10 of current value
if (less than full market share and
  inventory < 1/4 capacity)
    displacement = price diff from mean/mean
    Translate belief range upwards by displacement
elseif offer price > trade price
    Translate belief range downwards by 110% of overbid
elseif supply > demand and offer > historical mean price
    Translate belief range downwards by 110% of overbid
elseif demand > supply
    Translate belief range upwards by
    1/5 historical mean price
else
    Translate belief range downwards by
    1/5 historical mean price
```

```
PRICE-UPDATE-FROM-ASK(Commodity)
weight = percent of order unfilled
displacement = weight * price mean
if No units sold
    Translate believe range downwards by
    1/6 displacement
elseif Have less than 75% of market share
    Translate believe range downwards
    by 1/7 displacement
elseif offer price < trade price
    Translate belief range upwards by 120% of
    weight * overbid
elseif demand > supply
    Translate belief range upwards by 1/5
    historical mean price
else
    Translate belief range downwards by 1/5
    historical mean price
```

IV. EXPERIMENTAL PROCEDURE

Experiments were performed for both the general example and a large number of examples using random production rules. We created simulators that allow computer controlled trading agents to buy, sell, produce, and consume commodities. These agents were assigned roles so that an equal number of agents were initially in each role. Between 1000 and 10000 rounds of simulation were performed, during which time each agent interacts with other agents either as a buyer or a seller, and unsuccessful agents are replaced by new agents in different roles.

The individual trading agents were designed to provide realistic behavior by maintaining unique state that is updated based on their individual experiences. We required agents to be partitioned into classes based on role to take into account for the human tendency to specialize labor. In the real world an individual typically holds one job at a time, although the individual may change jobs over time. We restricted agents from trading in commodities that they did not require for production, nor produce. This is not a significant restriction as a rule could be added that converted a commodity into itself, making an agent technically a producer without performing any real production. For example, a rule that transforms one unit of commodity A into one unit of commodity A would allow the agent to buy and sell this commodity, without changing the amount present in the world. This rule would allow an agent to speculate in commodity A. An assumption that an agent will act to maximize profits with only knowledge that it personally gained indicates that agents will act in a rational and fair manner.

The following assumptions are made regarding the trading agents:

- Traders are heterogeneous, having unique pricing beliefs, roles, inventories, and money on hand.
- Traders follow role-specific rules for consuming and producing commodities.
- Traders use an arbitrary unit of currency as a standard for pricing commodities.
- Traders only trade in commodities that they personally produce or consume.
- Traders are allowed to maintain a limited inventory of each commodity.
- Traders act to maximize their long-term profits.
- Traders do not have perfect knowledge of the market.
- Traders learn from personal experience.

The random number generator was assigned a unique seed for each run. The use of random numbers to determine prices within a confidence interval, or to determine if an unexpected event occurs caused the simulator to produce different results, but each similar in the general behavior. As we will discuss in a later section, our simulation exhibits a chaotic sensitivity to small changes in the initial conditions.

In each round of simulation each agent performs a production operation, generates offers to buy or sell certain commodities, and delivers these offers to the auction house. The central auction house collects these offers and stores them in separate offer books (one book for bids, one book for asks). Once all agents have had an opportunity to enter their offers, the auction house resolves the trades using a distributed double auction, as described by Steiglitz, Honig, and Cohen [6].

SIMULATION-LOOP

```
for round = 1 to number-of-rounds
    for each trading agent
        perform production
        generate offers
    resolve offers
```

Production can be generalized as the conversion of one set of commodities (referred to as a *basket*) into another basket of commodities. The details of how this is performed is implementation dependent, but in general one verifies that the necessary materials are on hand, removes these from an agent's inventory, and adds the production product to the agent's inventory. We therefore assume that each agent maintains a separate inventory capable of holding an arbitrary number of each commodity. In practice a game may place limitations on the size of this inventory.

An agent creates offers by examining all commodities that it either consumes or produces. If the agent is running low on a commodity that it consumes, CREATE-BID is called to create an offer to buy an appropriate amount of this commodity. The offer is then sent to the clearing house where it is added to the bid book for that commodity. Similarly, if an agent has produced some commodities that it does not need, CREATE-ASK is called to create an offer to sell an appropriate amount of this commodity. This offer is sent to the clearing house and is added to the ask book for that commodity.

Once each agent has had the opportunity to add a set of offers to the appropriate offer books, the offer books are shuffled to remove any bias due to the order the agents were processed, and both books are sorted by price. The central clearing house will then use a double-auction to determine the resolution of each of these offers.

We are interested in the quality of the simulations only to the extent that it allows us to provide prices that appear reasonable to players. So while we have no strict need for high quality results, we sought techniques that were fast and gave us good behavior. Double auctions were selected both for their efficiency, and their ability to approximate theoretically predicted behaviors (see, for example, Smith [3], and Gode and Sunder [21]). The use of these auctions in experimental economics for the past fifty years gives us confidence that they represent a sound technique.

In this type of auction, RESOLVE-OFFERS matches the highest bid with the lowest ask, a trade occurs, the offers are updated to reflect the quantity of a commodity exchanged, and any offers with zero units unfilled are removed from the book. This process continues until either the bid or asks book is emptied. The offer books are shuffled at the beginning of a round to eliminate bias among agents with the same offer price. Note that when this matching stops one of the books will likely have offers remaining, and these are reported to the issuing agent as being rejected. During offer resolution, the minimum of the bid and ask quantities are exchanged at the average of the bid and ask price as discussed by Nicaolaisen *et al.* [5]. Inventories of each agent are adjusted by the amount of the trade, and the amount of currency each agent has is also adjusted.

```
RESOLVE-OFFERS (Commodity)
Shuffle both bid and ask books for Commodity
Sort bid book in order of decreasing offer price
Sort ask book in order of increasing offer price
while both books are non-empty
    buyer = the first bid in the book
    seller = the first ask in the book
    quantity traded =
    Min(units offered by seller,
        units desired by buyer)
    clearing price =
    Average ( seller's offer price,
        buyer's offer price )
    if quantity traded > 0
        reduce units offered by seller by
            quantity traded
        reduce units desired by buyer by
            quantity traded
        transfer quantity traded units of
            Commodity from seller to buyer
        transfer clearing price *
            quantity traded from buyer to seller
        both seller and buyer update
        their price model
    if quantity offered by seller = 0
        remove the first ask from the book
    if quantity desired by buyer = 0
        remove the first bid from the book
Remaining offers are rejected and the issuing
agent updates its price belief
```

At the end of each round, agents are notified of the quantity of commodity traded as a result of their offer. This notification contains market statistics for the current round, such as trade volume, the average price for trades, the high and the low price for the commodity in the offer. Agents will then update their personal price models to reflect their belief in the true value of this commodity. Note that there is no single true value for a commodity, but rather a set of beliefs held by each agent that trades in a commodity. Over time it is observed that the agents converge to a single shared belief in a commodity's value, although external events (shocks) can cause the market to shift to a new shared belief.

Although individual agents in our world maintain no personal history, the clearing house does maintain some historical information that is available to all traders in the market. Agents are therefore required to adjust their personal beliefs about the value of a commodity based on this public information and any information privately learned from prior offer resolution. This public information consists of:

- The average price for each commodity within some user-defined window.
- The average quantity of each commodity offered for sale within some user-defined window.
- The average quantity of each commodity bid on within some user-defined window.

V. Agent Replacement

Ideally, agents in a game should change roles when there is economic pressure to do so. If we treat these roles as professions, we may evaluate an agent's performance by their profitability, and decide to change jobs when the agent is no longer profitable. These changes in roles is necessary to provide variations in supply and demand for commodities, as commodities high in demand will attract new agents and in turn increase the available supply. Our experiments have shown that the simulated markets move towards a set of agents supportable under the current economic conditions, and reallocate agents when market conditions change. In practice, simulated market conditions are constantly changing so the market never converges to a stable set. This automatic reallocation of agents is a benefit to the game designer, as it allows adjustments in the population of NPCs without explicitly coding for causative events. For example, an interruption in supply for a commodity (such as timber) will affect industries directly dependent on the commodity (shipbuilders for example) as well as indirectly (farmers who provide food to the shipbuilders). These external events may, depending on the magnitude and duration, cause agents to go bankrupt.

An agent that is unable to remain profitable will eventually go bankrupt, and be replaced with a new agent of the currently most profitable type. This profitability statistic is a moving average of profits for a particular type of agent over some user-define number of prior rounds, ensuring that recent performance is evaluated. We have seen good results with windows between 8 and 15 rounds, but a particular set of production rules may work better with other values. It is a reasonable assumption that a recently bankrupt agent was not in a profession that was doing well, and therefore this replacement strategy acts to maintain a constant population size but varies the composition of agent types. As a result, as long as bankruptcies occur, the simulation will make adjustments to the distribution of agent types. Ideally, absent of some external disruption, there will be a point where no future bankruptcies occur, as the market is capable of supporting each agent indefinitely.

Our results are consistent with accepted economic theory. Adam Smith [2] theorized that people trading in an open market would lead to the production of the proper quantities of commodities and the division of labor. Our results support this belief, since agents that are not profitable are bankrupted and replaced by more profitable agents.

The first fundamental theorem of welfare economics states that a market with a supply/demand equilibrium leads to a Pareto-efficient allocation of resources, meaning that no change to the resource allocation can be made without making at least one trader worse off [22]. This would suggest that when the market is in equilibrium the allocation of agents to roles will over time tend to an optimal value [23]. In practice no market ever moves into equilibrium, but instead will move into a neighborhood that is near equilibrium and oscillate about the equilibrium point [3].

Ohe economic system will constantly attempt to determine a distribution of agent roles that results in market equilibrium, however it never reaches this goal and instead orbits around equilibria until market conditions establish new equilibria points. At this time, the agent distribution is seen to adjust and move towards these new equilibria.

VI. A General Example

Our simulation used various techniques to exercise this economic system. The most general form of production was to allow the simulator to call agent-specific routines that would update inventory. This allowed us to implement complex production rules without restriction, while updating the price models in a manner consistent with an actual game. One such ruleset allowed agents to be farmers, miners, refiners, woodcutters, or blacksmiths. These agents produced, and consumed food, ore, wood, metal, and tools according to the production rules defined by Farmer-Production, Miner-Production, Refiner-Production, Woodcutter-Production, and Blacksmith-Production.

This example was created to illustrate a typical economy, as might be found in a fantasy role playing game. As the rules were implemented using arbitrary code, the designer is free to create as complex a ruleset as desired.

Due to space limitations we provide a single representative rule for reference, an extended version of this paper with the entire example is available from the authors.

```
Farmer-Production
if has-wood and has-tools
    produce 4 units of food
    consume 1 unit of wood
    break tools with prob 0.1
elseif has-wood and has-no-tools
    produce 2 units of food
    consume 1 unit of wood
else
    agent is fined $2 for being idle
```

VII. Random Generation of Production Rules

In addition to testing with the production rules described above we have developed a second simulator that creates random production rules, in order to demonstrate that our results are not dependent on any single set of rules. To achieve this we expressed the production rules in a matrix format, allowing us to assign random values to the matrix and then simulate a set of agents operating under these rules. We claim that if we observe acceptable behavior from economies using randomly generated rules, then we have a suitably general solution that will perform well with rules that a designer might select. We do not claim that all rules will perform well, only that a large set will. In particular rules that express a non-closed economy (where agents consume a commodity that is not produced) are not going to produce pleasing results under any economic system.

The matrix form for production rules defines a rule as a set of commodities that is converted into another set under a probability distribution. For example, the rule shown in Equation (1) allows an agent to convert two units of $Commodity_1$ into one unit of $Commodity_4$. Additionally, the agent is required to possess one unit of $Commodity_3$ that is consumed 10% of the time. In a simulation round, an agent is permitted to perform production using one of these rules. If the agent does not possess inventory listed on the left-hand side of a rule, the production is not allowed and the agent must consider other rules. It is therefore possible for an agent to be unable to perform production in a given round due to inadequate inventory. In this situation we assess an idleness tax, to ensure that non-productive agents were eventually driven bankrupt. As each type of agent was allowed to select among several rules, we ranked the production rules in order of preference and had agents use the first rule in their set that they were able to execute.

This use of multiple production rules for an agent-type along with probabilities for terms in production rules allows us to model complex behavior including conditionals (such as, does the agent possess a tool or catalyst represented by $Commodity_3$ in our example).

$$\begin{gathered} 2 * Commodity_1 + Commodity_3 \Rightarrow \\ Commodity_4 + Commodity_3 (p = 0.9) \end{gathered} \tag{1}$$

Since these simulations were requiring large numbers of calls to the random number generator, we were concerned that we might exceed the default random number generator's period and bias our results. We replaced this generator with a Mersenne Twister random number generator (MT19937), which has a period of $2^{19937} - 1$.

Due to the large number of random experiments we performed, we were unable to study all of the results. We therefore established screening criteria to filter out unacceptable results, with the intention of counting the number of simulation runs that were well behaved. We arbitrarily selected a set of desirable features for a price graph, and then modified the filters until we were seeing only these types of graphs. Many of these features were based on the belief that we need to observe regular trades for each commodity if we are to judge the market's overall performance. We further wished to see that prices change over time, but wanted some long-term stability in prices. We define stability as the tendency of prices to return to equilibrium, as opposed to diverging to 0 or infinity. The exact values in the filters were therefore determined empirically from a representitive set of price graphs. The final criteria used were as follows:

- Each commodity was produced by at least one type of agent.
- Each commodity was consumed by at least one type of agent.
- No commodity goes more than 20% of the total number of rounds without trading.
- The average trade volume for a commodity is greater than one unit per round.
- The variance in commodity price is between 0.025 and 7.5 times the largest trading price.
- The average change in price is between 0.02 and 0.9, the variance of this change is also between 0.02 and 0.9.
- Fewer than 2 price inflections occur per round on average.
- The variance in the time between price inflections is less than 1.2 times the number of rounds.

We are therefore comparing both the variance in price and the variance in the first derivative of the price.

VIII. Experimental Results and Evaluation

When evaluating the performance of a complex system, one realizes that combinatorial growth in the number of possible interactions makes exhaustive analysis intractable. In our economic simulation, however, there are a limited number of ways in which a trader (human or agent) can affect the economy. Buying a large quantity of a commodity can reduce available supplies for other traders, and encourage agents to switch production to this commodity. Selling a large quantity of a commodity can increase supplies, and discourage agents from producing more of this commodity. And finally, we assume that agents have the ability outside of the market to interfere with supply and demand, by blockading an area or destroying resources. Our prime concern is with showing that our economic model recovers from even extreme behaviors, and therefore that the range of possible trader behaviors will not destabilize the system in the long term.

A. A Representative Ruleset

We first consider the general fantasy-RPG themed example discussed in Section VI. We feel this ruleset is representative of the types of rules one might use within a game.

Figure 2 shows the behavior of a simulation over 2000 rounds. If we arbitrarily decide that each round represents one day's activity in game time, we present almost 5.5 years of price data. A graph of the supply/demand ratio over time is shown in Figure 4, and demonstrates that the economy undergoes the full range of supply and demand imbalances. A concerted effort by multiple players could artificially bring about such an imbalance, but we see no long-term effects. Figure 2 demonstrates that the economy recovers once an imbalance is eliminated. This auto-recovery is necessary for long-term stability, and in particular eliminates the need for human intervention.

When looking at this long-term behavior, we observe that trading occurs within a bounded range, suggesting that prices are orbiting a stable equilibrium rather than diverging to either 0 or infinity. The system is constantly attempting to move back into equilibrium, while it is undergoing further perturbation as the result of trades. We consider the graph in Figure 2 to show long-term stability in prices as the system recovers from disruptive events on its own. We also note that prices are not precisely predictable, although some relationships can be seen over time. A close up of one part of the graph is shown in Figure 3. There is a correlation between commodities that are dependent on one another, as the prices of products move with the price of the raw materials. For example, the price of refined metal (shown with a dash-dot pattern) tends to rise and fall with the price of unrefined ore (drawn with dashes). The magnitude of the changes differs, but the local maxima at rounds 1020, 1060, 1110, 1140, 1170 and 1200 occur in both graphs.

Paradoxically the price of refined metal in Figure 3 appears to increase before the price of refined ore (the precursor product). It should be kept in mind that these figures show average prices over time, and an agent may choose to raise its offering price the moment it experiences resistance to a price, even if the market average does not yet reflect the belief in a price change. While we see correlations in prices, we also see the independent movement of prices. The prices of ore and metal in this case are not translations of each other, but vary within a limited range. This long-term stability is desirable, as it shows that the system does not undergo runaway inflation, but instead self-corrects.

The sensibility of prices is a subjective measurement, but as long as our simulated agents behave rationally we must accept that the prices they trade at make sense. We note in particular that as the prices of raw materials go up, the prices of finished goods increases with a slight delay as inventories are used up. Allowing an agent to stockpile a certain amount of a commodity provides a short-term buffer against price changes, and tends to dampen price swings. An internal consistency in prices occurs since the economy is a closed system, and each transaction influences future transactions. This consistency is predicted by accepted economic theory, and to the extent that our results agree with theory we are able to claim that our system's behavior makes sense.

B. Applications of Economic Theory

Proponents of General Equilibrium Theory believe that supply and demand will equalize over time, however our results do not support this. In particular Arrow and Debreu [24] argued in favor of this equalization, assuming that traders in the market had

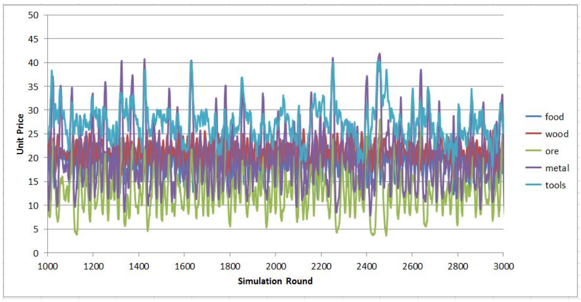

Fig. 2. The system has stable long-term behavior, moving in and out of equilibrium.

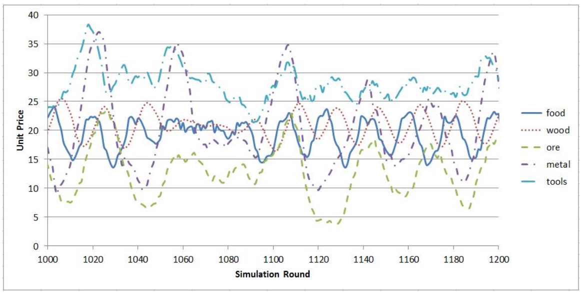

Fig. 3. Detail of the previous figure, showing similar behaviors among commodities.

perfect information and responded instantly to market changes. Their argument is intuitive when one considers that an imbalance in the supply/demand ratio should result in price changes that result in changes to supply and demand and return the system to equilibrium. However in the real world Arrow and Debreu's constraints do not hold, nor do they hold in our simulation. Traders create offers based on their belief in the market price, but without knowing the beliefs of other individuals. Traders are able to estimate the beliefs of others based on their observations of trades that complete, but only have perfect information on their own trades and their own price beliefs. Traders also do not respond instantly to market changes, as they only update beliefs after they have tendered an offer and seen how it was received. This delay, coupled with the time needed for market averages to converge following a shift in belief cause the agent

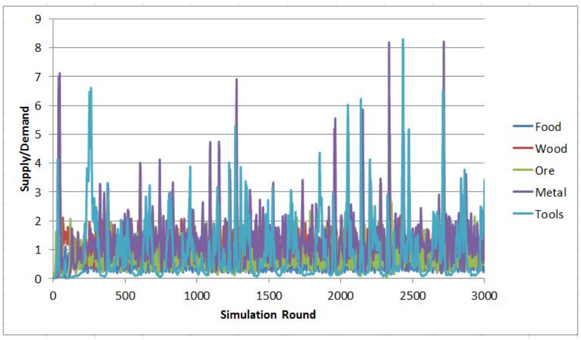

Fig. 4. Long-term supply/demand behavior, showing the system in a variety of regimes.

to respond slowly to market changes. We believe this is a useful property, as it prevents agents from overreacting to short-term market changes, as well as better reflecting how a trader in the real world would respond.

We look to Vernon Smith [3], a pioneer in the field of experimental economics for an explanation. Smith explains that supply and demand can only set broad limits on the behavior of the market, as any successful trades remove a quantity of supply and demand from the market and therefore alter the supply and demand curves. In a later paper Smith [4] explains that "all information on the economic environment is private; far from having perfect or common information" and "prices and allocations converge quickly to the neighborhood of the predicted rational expectations competitive equilibrium". So in the ideal case supply and demand would converge to the same value, in real experiments they will only be in the same neighborhood. This agrees with our observations of the supply/demand ratio over time.

In one experiment 500 heterogeneous agents were simulated for 10000 rounds of trading, and the supply/demand ratios were graphed over time. These ratios were not constant, but instead varies between approximately 0.5 and 2, repeatedly crossing the line $y = 1$ (representing the equivalence of supply and demand). We conclude that the market is constantly trying to make these values equivalent, but overshooting and then correcting itself. As this behavior agrees with Smith's observations, our confidence in our results is further strengthened.

We have observed that agent profitability tends to zero over time, as prices for raw materials increase to the point where buyers refuse to bid on them. Adam Smith [2] discusses a similar phenomenon in the Wealth of Nations (Chapter 10, Part II) where he notes that the landlord will raise prices until the tenant is left with "the smallest share with which the tenant can content himself without being a loser, and the landlord seldom means to leave him any more". If we look at the average agent profit (by type) over time in Figure 6 we see profitability orbiting the zero equilibrium, the disruption due to the external event, and the recovery as profits again trend towards zero. While the external event does create a significant disruption, once the event completes the system returns to orbiting the equilibrium. The long-term behavior of our simulated economies therefore agrees with accepted economic theory in this aspect.

C. Response to Extreme Stimuli

Our system is able to adapt and recover from external perturbations and shocks. This is a useful feature since it addresses the type of market manipulation that players might choose to engage in. Figure 5 shows the effect of a short term interruption in the supply of wood. Between round 600 and round 700 woodcutters were unable to harvest wood, simulating a forest fire that has eliminated the resource. We see the price

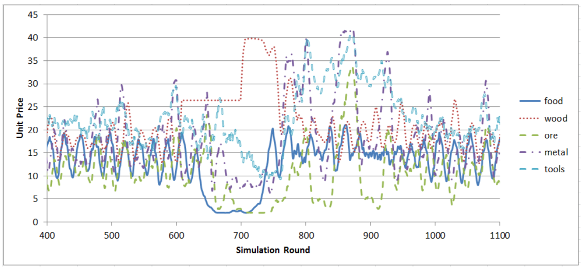

Fig. 5. An interruption in wood supply and subsequent recovery.

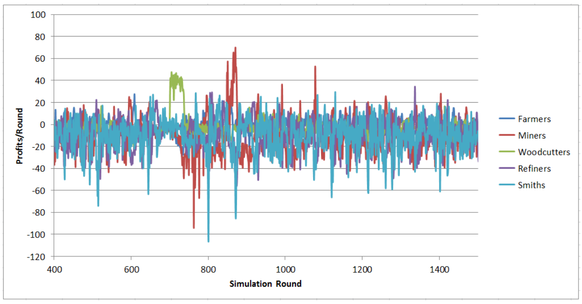

Fig. 6. Agent profitability by type for the forest-fire scenario, demonstrating an orbit about the equilibrium.

of wood remain fixed during this period, as there are no sales there will be no price updates (a price is only established when an offer is accepted). The prices of ore and tools begin to rise during this period as both products use wood (to shore up mine shafts and for tool handles) and existing wood inventories are depleted. While wood production resumes on round 700, prices continue to rise for another 50 rounds as the unsatisfied demand greatly exceeds the limited supply. The prices of products that depend even indirectly on wood increase, although there is a delay before the market adjusts these prices. In this case the delay is due to accumulated inventories being drawn down, and acting as a moderating force on prices. These prices will eventually resume their previous pricing behavior, but there is a time lag as inventories are depleted and agents start to believe that wood is no longer scarce. Even a serious shock to the economy such as the fire creates no long-term harm, as eventually we observe the system returning to equilibrium at approximately round 950.

One should keep in mind when looking at these graphs that the prices are partially a function of random chance. Tools break at random times, and agents enter bids based on random guesses within their price confidence interval. As a result there can be large price fluctuations if enough of these random events occur in a short interval. This can be a good thing for both the designer and the player, as it means that one may never exactly predict price behavior. However, the overall price trends do follow patterns, and do react to major events (such as the forest fire in Figure 5). It should be possible for players to engage in arbitrage if they so choose. A knowledgeable player who becomes aware of the lack of wood could buy up tools and ore and wait for the market prices to increase, then sell them at a profit. Short term price shocks are therefore not a problem as long as the long-term behavior of the economy is consistent.

The allocation of roles to agents is an important product of this model, as it is necessary to know how many agents may be supported in a role. If we consider a community with N agents and M roles, one might need to know how to partition the N agents into M sets such that the agents remain profitable. This type of question comes up when we consider adding a new NPC (the N+1th agent) to the game and wish to know what role this NPC should take on. A village populated entirely by woodcutters would raise the question of how these woodcutters find food, or where all of this wood is going. One may avoid these situations by ensuring that all population distributions are viable, that there is a need for each of the agents and that each of the agent's needs are met. We obtain this information from the simulation at minimal cost by observing the profitability of roles over time.

At any point in time there will be an ordering of the M roles by profitability. The more demand there is for a role, the more profit one will expect for agents in this role. This is the result of high demand driving up prices for the products of this role faster than the materials needed by agents in this role. Conversely agents in a role that is not in demand will find it difficult to execute trades, yet will still have overhead (food in this example). When a new agent is added to the simulation, if we bias the role selection by the profitability of the roles, we ensure that high demand roles gain agents and low demand roles lose agents. At any point in time the population of agents represents a viable community, since agents will be removed when they are no longer able to provide for themselves. Figure 7 shows changes in the distribution of agents by role over time, starting from an arbitrary (and unsustainable) distribution. In this example farmers and woodcutters are in high demand since both provide resources needed by other agents, so it makes sense that new agents would favor these roles. The exact number of agents at a point in time is a function of random events, but also of the ruleset and ontology.

Short-term random events prevent the market's role allocation from reaching equilibrium; however there are stable patterns observed. We also observe that roles producing commodities that are high in demand will have more agents than those producing less needed commodities. As supply and demand change over time, the need for certain roles changes over time, and the market moves towards a different allocation. At each point in time, the number of agents in a particular role approximates the number of profitable agents under current market conditions. As a result, a census of agents in the market allows us to determine a reasonable distribution of agent types, and we are therefore able to create a community of N agents and know that the market will reallocate the roles until an acceptable distribution is found. Furthermore we are

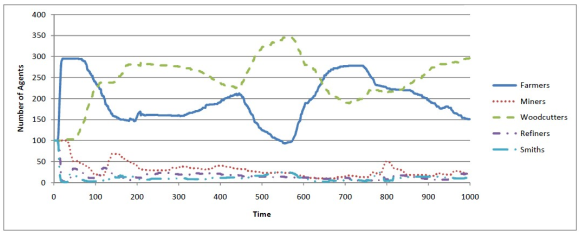

Fig. 7. The distribution of agent roles over time.

able to start the simulation with an arbitrarily chosen distribution (we initially assigned equal numbers of agents to each role), and know that the system will quickly reallocate these agents into more appropriate roles.

D. Experiments with Random Economies

In addition to our experiments with the general example, we performed a large number of experiments using random economic rules. As discussed earlier, we applied a filtering function to test price graph characteristics and decide if a particular set of rules produced acceptable results. We used UNT LARC's cluster of PS3 consoles to evaluate these random economies, with the simulation performed on the Cell processors' SPE units. The high degree of parallelism and the high performance of these Cell processors allowed us to evaluate millions of distinct rule sets in a few hours. We tested approximately 1.5 billion random economies and found 2.7 million (0.19%) that passed all of the filter criteria. These economies appear to be uniformly distributed, as we observed roughly the same fraction for multiple smaller runs using different random number seeds. We calculated that there were approximately 2^{160} distinct matrices, and assuming that 0.19% of these have acceptable performance we have more than 2^{150} economies to choose from. Figure 8 shows two representative price graphs, demonstrating price behavior under random rules. Systems using random rules are less likely to be well behaved than those a designer might create. Based on the large number of well-behaved random economies, we believe that the behavior of this economic system is independent of any single ruleset, and we are confident that a designer would be able to create a ruleset with similar performance.

The calculations required to update the economy can be carried out very quickly. A single 3.2GHz SPE can perform approximately 200000 agent updates per second, including simulator overhead such as data logging. Most games do not have this many NPCs, nor do they require them to be constantly buying and selling. Only two floating point values are required per commodity per agent, which we feel to be particularly light. We are confident that adding an economic simulation to a game will not add a significant burden in terms of either processing time or memory requirement, and as a result this technique is feasible.

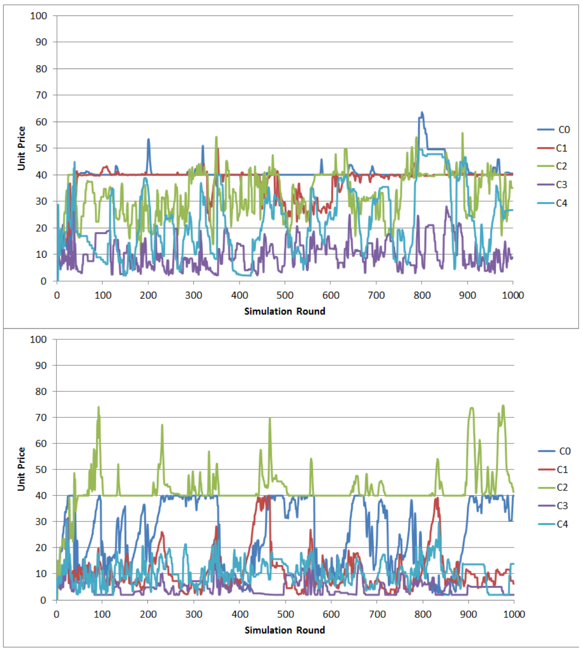

Fig. 8. Two random economies

E. Observed Chaos

Chaos is defined as sensitivity to initial conditions affecting the outcome, and this describes behavior seen in our simulation. We first became aware of the issue when we observed differences in the output when the compiler's optimizer was turned on. Our investigation of this phenomena concluded that even though we used double-precision floating point in our implementation, discretization errors were present in our intermediate results due to the inability of the compiler to express certain floating point values (such as 0.1) as exact values in binary. Furthermore, the optimizer was reordering floating point operations, causing these discretization errors to propagate differently than they would in an unoptimized version.

The magnitude of the sensitivity was demonstrated by a one-time addition of 10^{-9} units of currency to a single agent during a simulation. We observed that several agents went bankrupt who would otherwise have remained solvent. In addition, after 500 rounds of simulation the price of certain commodities varied by around 20% from the normal runs. This magnitude of error is within the observed discretization error for 0.1, and can be expected to occur normally during simulation when calculating moving averages.

This chaotic behavior is not an error, but can be expected in an iterative simulation that employs feedback. Our model was tested with production rules that coupled different agent types. This means that each type of agent produces a product needed by at least one other type of agent, and uses a product produced by at least one other type of agent. As a result of this coupling, any perturbation of one agent will propagate to other agents. Furthermore, an amplification effect occurs as a result of continuous errors becoming discrete errors. Say that one agent is saving money to buy a needed tool, and in one case the agent comes within ϵ of being able to afford the tool before going bankrupt. In another run, due to chaotic effects, this same agent may gain an extra ϵ of currency and buy the tool. And as a result of owning this tool, the agent may become profitable and remain in business. As a result, this agent continues to have an impact on the economy, buying and selling goods and affecting prices on these goods. A small error of ϵ has a much larger affect, once it results in a discrete change (the number of tools owned by the agent changing from 0 to 1).

We believe that this behavior is desirable, since it makes the impact of player actions harder to predict. It is important to note that the changes to the economy outside of the ϵ error are justifiable under the production rules.

Oxley and George [25] note that economics can be chaotic. Rosser [26] also gives a good explanation of economics as a complex dynamic system. Our model does indeed have the following characteristics found in chaotic complex systems (see Arthur, Durlauf, and Lane [27]):

- Disperse interaction: Agents interact with a subset of other agents
- No global controller: No single agent may control the market
- Tangled interactions: The production models are usually coupled.
- Continual adaptation: Agents constantly update their beliefs about prices
- Perpetual novelty: In a chaotic phase, markets are created and destroyed as the agent mix stabilizes. Also until agents' beliefs in commodity prices converge commodities will frequently trade at prices that contradict these beliefs.
- Out of equilibrium dynamics: Prices may not move to an attractor, but may orbit perpetually.

IX. Conclusion and Future Work

We have presented a technique for simulating a game economy, resulting in changing prices, trade volumes, and a distribution of roles within the economy. We have shown that our economic system produces reasonable prices for arbitrary sets of commodities and agent types, and the experiments with random economies demonstrates that performance is independent of any particular ontology. We have demonstrated that our system adapts to and recovers from external shocks, and that the system returns to the neighborhood of market equilibrium after the shock has abated. We also saw how the simulation was able to assign roles to individual agents, and modify the distribution of roles as changes in market conditions warranted. Finally, our analysis of the time and memory requirements for this simulation suggests that this system is feasible for use in computer games, where machine resources are often at a premium.

Game economic models do not have the same requirements as traditional economic models, and should therefore be evaluated differently. Assessing the accuracy of our model is difficult, as we are not able to compare prices and trade volumes to those from an actual economy. However, accuracy is less important than demonstrating interesting behavior. To be interesting, we believe that prices should be reactive, consistent with events, and have enough variability that players may observe changes over time.

The behavior of prices and agent roles appears to be of acceptable quality, as they demonstrate constant small magnitude changes and yet respond to significant events in the market with larger changes. As prices are a function of supply and demand, any event that alters either of these values will result in a corresponding change to prices. If a player is able to set a forest on fire, they will see the price of wood change. This type of reactivity is consistent with player expectations, and we believe sufficient to allow players to suspend their disbelief.

Trade volumes were observed to update in response to aggregate supply and demand by individual agents. The quantities of a product available for purchase depend on how profitable it is to produce this product, and how much competition there is to buy this commodity. During the forest fire scenario we observed trade volumes decrease as existing inventories were depleted, prices rose as the competition for the shrinking inventory increased, eventually the supply disappeared entirely and the demand began to disappear as it was no longer profitable to be in a role that required raw materials that were not available. Our model appears to be sufficiently reactive, and the price updates are consistent with changes in supply and demand.

In the future, we hope to investigate using this economic data to create towns and villages populated by NPCs. The statistical techniques used by pencil and paper games tend to create communities that look very similar, and we hope that we are able to use simulated economic data to introduce variety into the world while preserving believability. In particular, deciding how many people should live in an area, and how they support themselves is a difficult task. A simulation, such as we provide, could be an invaluable asset for automating this sort of content creation.

It would be useful to expand the simulation from a single market to a set of markets interconnected by slow trade routes, and then introduce regional resources. We wonder if regional markets would converge to a common set of prices, or would remain distinct in their beliefs. The speed and cost of transportation likely will play a major role in this behavior, as an infinitely fast and free transportation network would reduce problem to the single market discussed in this paper.

We leave open the problem of measuring the extent to which a dynamic economy modifies or improves the user experience in an RPG. This would benefit most from researchers with a strong social sciences background.

REFERENCES

[1] V. Pareto, “The New Theories of Economics,” *The Journal of Political Economy*, pp. 485–502, 1897.

[2] A. Smith, *An Inquiry Into the Nature and Causes of the Wealth of Nations*. Adam and Charles Black, 1863.

[3] V. Smith, “An Experimental Study of Competitive Market Behavior,” *The Journal of Political Economy*, vol. 70, no. 2, pp. 111–137, 1962.

[4] ——, “Rational Choice: The Contrast Between Economics and Psychology,” *Journal of Political Economy*, vol. 99, no. 4, pp. 877–897, 1991.

[5] J. Nicolaisen, V. Petrov, and L. Tesfatsion, “Market Power And Efficiency In A Computational Electricity Market With Discriminatory Double-Auction Pricing,” *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 5, pp. 504–523, 2001.

[6] K. Steiglitz, M. Honig, and L. Cohen, “A Computational Market Model Based on Individual Action,” *Market-Based Control*, 1996.

[7] M. Benisch, A. Greenwald, I. Grypari, R. Lederman, V. Naroditskiy, and M. Tschantz, “Botticelli: A Supply Chain Management Agent Designed to Optimize Under Uncertainty,” *ACM Transactions on Computational Logic*, vol. 4, no. 3, pp. 29–37, 2004.

[8] K. Chatzidimitriou, A. Symeonidis, I. Kontogounis, and P. Mitkas, “Agent Mertacor: A Robust Design For Dealing With Uncertainty And Variation In SCM Environments,” *Expert Systems with Applications*, vol. 35, no. 3, pp. 591–603, 2008.

[9] W. Ketter, E. Kryzhnyaya, S. Damer, C. McMillen, A. Agovic, J. Collins, and M. Gini, “MinneTAC Sales Strategies For Supply Chain TAC,” in *Proc. of the Third Intl Conf. on Autonomous Agents and Multi-Agent Systems*. New York, New York: ACM, July 2004, pp. 1372–1373.

[10] D. Pardoe and P. Stone, “Adapting In Agent-Based Markets: A Study From TAC-SCM,” in *Proceedings Of The 6th International Joint Conference On Autonomous Agents And Multiagent Systems*. Honolulu, Hawai’i: ACM, May 2007, pp. 677–679.

[11] A. Roth and I. Erev, “Learning in Extensive-Form Games: Experimental Data and Simple Dynamic Models in the Intermediate Term,” *Games and Economic Behavior*, vol. 8, no. 1, pp. 164–212, 1995.

[12] C. Flores and L. Garrido, “Learning of Market Prices Based on Temporal Differences for a TAC-SCM Agent,” in *Proceedings of the 2008 Seventh Mexican International Conference on Artificial Intelligence*. IEEE Computer Society, 2008, pp. 414–419.

[13] F. A. Dahl, “A Reinforcement Learning Algorithm Applied to Simplified Two-Player Texas Hold’em Poker,” in *Proceedings of the 12th European Conference on Machine Learning (ECML-01)*. Springer, 2001, pp. 85–96.

[14] D. Pardoe and P. Stone, “Bidding For Customer Orders In TAC SCM,” in *AAMAS 2004 Workshop on Trading Agent Design and Analysis*. Springer, 2004, pp. 143–157.

[15] R. Schapire, P. Stone, D. McAllester, M. Littman, and J. Csirik, “Modeling Auction Price Uncertainty Using Boosting-Based Conditional Density Estimation,” in *Proceedings of the Nineteenth Intl. Conference on Machine Learning*, 2002, pp. 546–553.

[16] M. Wellman, J. Estelle, S. Singh, Y. Vorobeychik, C. Kiekintveld, and V. Soni, “Strategic Interactions In A Supply Chain Game,” *Computational Intelligence*, vol. 21, no. 1, pp. 1–26, 2005.

[17] W. Ketter, J. Collins, M. Gini, P. Schrater, and A. Gupta, “A Predictive Empirical Model For Pricing And Resource Allocation Decisions,” in *Proceedings Of The Ninth International Conference On Electronic Commerce*. Minneapolis: ACM, 2007, pp. 449–458.

[18] Z. Simpson, “The In-Game Economics of Ultima Online.” San Jose: Game Developers Conference, 2000.

[19] E. Castronova, “Virtual Worlds: A First-Hand Account of Market and Society on The Cyberian Frontier,” December 2001.

[20] D. Meadows, “Tools for Understanding the Limits to Growth: Comparing a Simulation And a Game,” *Simulation & Gaming*, vol. 32, no. 4, p. 522, 2001.

[21] D. Gode and S. Sunder, “Allocative Efficiency Of Markets With Zero-Intelligence Traders: Market As A Partial Substitute For Individual Rationality,” *The Journal of Political Economy*, vol. 101, no. 1, pp. 119–137, 1993.

[22] K. Arrow, “The Organization of Economic Activity: Issues Pertinent to the Choice of Market versus Nonmarket Allocation,” *The analysis and evaluation of public expenditure: The PPB System*, vol. 1, pp. 59–73, 1969.

[23] L. Makowski and J. Ostroy, “Appropriation and Efficiency: A Revision of the First Theorem of Welfare Economics,” *The American Economic Review*, vol. 85, no. 4, pp. 808–827, 1995.

[24] K. J. Arrow and G. Debreu, “Existence of an Equilibrium for a Competitive Economy,” *Econometrica*, vol. 22, no. 3, pp. 265–290, 1954. [Online]. Available: http://www.jstor.org/stable/1907353

[25] L. Oxley and D. George, “Economics on the Edge of Chaos: Some Pitfalls of Linearizing Complex Systems,” *Environmental Modelling & Software*, vol. 22, no. 5, pp. 580–589, 2007.

[26] J. Rosser Jr, “On The Complexities of Complex Economic Dynamics,” *The Journal of Economic Perspectives*, pp. 169–192, 1999.

[27] W. Arthur, S. Durlauf, and D. Lane, *The Economy as an Evolving Complex System II*. Addison-Wesley Reading, MA, 1997.

Jonathon Doran Jonathon Doran received the B.S. degree in Computer Science from the University of Colorado at Boulder, Colorado in 1998, and the M.S. degree in Computer Science from the University of North Texas in 2010. Currently he is working towards the Ph.D. degree in Computer Science and Engineering at the University of North Texas, Denton.

His research interests are procedural content generation, agent-based simulations, and AI techniques for role playing games.

Ian Parberry Ian Parberry received his PhD in Computer Science from the University of Warwick in England in 1984. He is currently a Professor in the department of Computer Science and Engineering at the University of North Texas. He is a pioneer of game programming in academia, having taught game programming classes since 1993. He is the author of seven books, four of them on game programming, and more than 70 articles on a wide range of computing subjects including (in addition to game programming) algorithms, complexity theory, parallel computing, and neural networks.