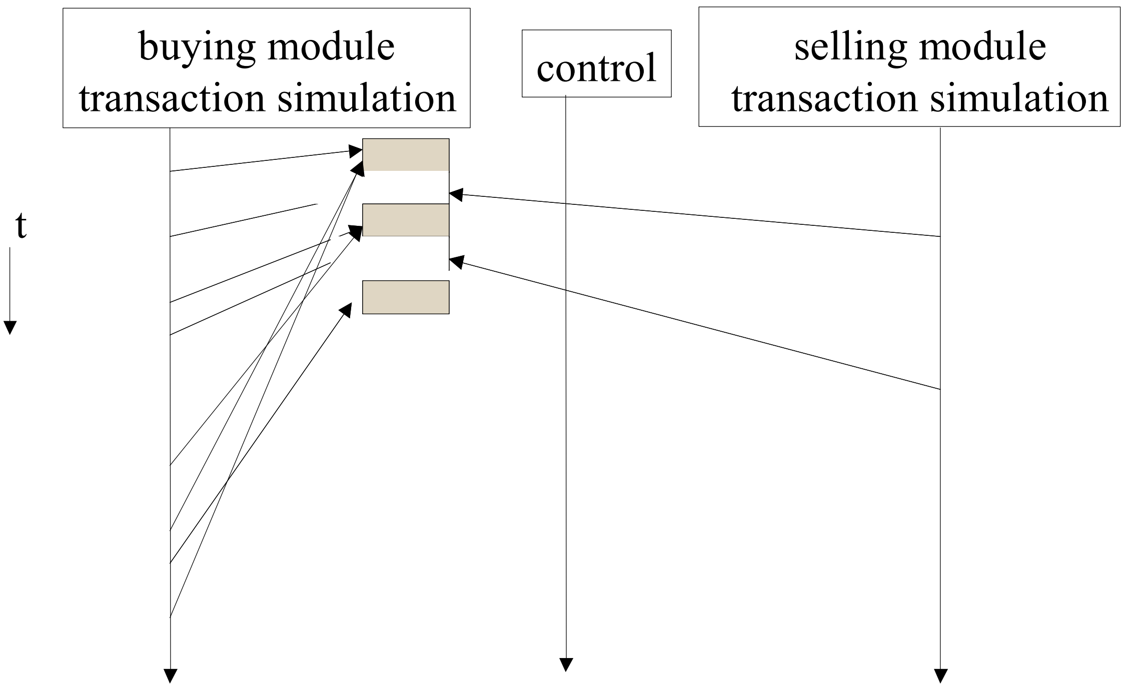


Dynamic submodel integration using an offer-accept discrete event simulation

John E. Abraham, University of Calgary
John Douglas Hunt, University of Calgary

Conference paper
Session XXX

Moving through nets:
The physical and social dimensions of travel
10th International Conference on Travel Behaviour Research
Lucerne, 10-15. August 2003

Dynamic submodel integration using an offer-accept discrete event simulation

John E. Abraham and John Douglas Hunt
Civil Engineering
University of Calgary
Calgary, AB, Canada

Phone: +1-403 230-5897 (JEA)
eMail: jabraham@ucalgary.ca jdhunt@ucalgary.ca

Abstract

Newer approaches for modelling travel behaviour require a new approach to integrated spatial economic modelling. Travel behaviour modelling is increasingly disaggregate, econometric, dynamic, and behavioural. A fully dynamic approach to urban system modelling is described, where interactions are characterized as two agents interacting through discrete events labelled as "offer" or "accept". This leads to a natural partition of an integrated urban model into submodels based on the category of what is being exchanged, the type of agent, and the time and place of interaction.

Where prices (or price-like signals such as congested travel times) exist to stimulate supply and/or to suppress demand, the dynamic change in prices can be represented either behaviourally, as individual agents adjust their expectations in response to their personal history and the history of the modelled region, or with an "auctioneer" from micro-economic theory, who adjusts average prices. When no auctioneers are used, the modelling system can use completely continuous representations of both time and space.

Two examples are shown. The first is a demonstration of a continuous-time continuous-space transaction simulation with simple agents representing businesses and households. The second shows how an existing model – the Oregon TLUMIP project for statewide land-use and transport modelling – can be adapted into the paradigm.

Keywords

Model Theoretical Framework, Market Treatments, Integrated Models, International Conference on Travel Behaviour Research, IATBR

Preferred citation

Abraham, John E. and John Douglas Hunt (2003) Dynamic submodel integration using an offer-accept discrete event simulation, paper presented at the 10th International Conference on Travel Behaviour Research, Lucerne, August 2003.

1. Introduction

1.1 Overview and Objectives

This paper concerns dynamic modelling of spatial economic systems. By *dynamic*, we mean that time is treated explicitly, and the simulation moves through time, with simulated events dependent on past history. By *spatial systems* we mean the study of systems where location matters – where transportation costs and the other aspects associated with distance cannot be assumed away. By *economic systems* we mean two things. First, we mean a system in which there are some scarce resources to be allocated. For these resources there is an element of competition or rationing, and there may be a price that emerges because of the interaction between supply and demand and the relationships between prices and costs and willingness to pay. Second, we mean a system where individuals and other entities (households, firms, establishments, government, etc.) interact with each other to satisfy each other's mutual wants and needs.

The paper presents a system for fully dynamic modelling that can incorporate:

1) infinite detail in time and space, so that predetermined "zones" and "time steps" are not imposed on behavioural algorithms.
2) arbitrary disaggregation of decision makers, to the level of detail that is desired

The system is well positioned to embrace newer advances in transportation modelling theory, which is increasingly based on econometrics and statistics, with disaggregate modelling of individuals, and an explicit representation of time and dynamics.

The system is a framework for representing *interactions*, and does not specify much about how to model the behaviour of individual agents. Specific attention is paid to "economic" interactions, where resources are scarce and must be allocated, and to free-market transactions, where goods and services are exchanged for money to each agent's mutual benefit. The emphasis on economic and free market interactions is because these are more complicated than many of the other transactions that occur.

We speak of "submodels", which are algorithms for the dynamic representation of certain decisions or decisions makers. Each submodel is implemented in computer code, usually developed by a closely knit team of scientists and computer experts. We do not wish to disrupt or disturb the work of these closely knit teams, only to integrate their work into dynamic models of spatial economic systems. Thus we describe these submodels as standalone units, and focus on how the framework can serve to connect them.

The paper grows out of research and experience with *land-use transport interaction* models. These models focus on supply and demand, and hence markets, in final and intermediate goods and services, floorspace, labour, retail goods, land and transportation infrastructure and services. The models are meant to be practical policy analysis models, to be used primarily within government agencies. They focus on a detailed representation of how transport and spatial policy and infrastructure influences the growth and evolution of the transportation system and the city or region that it supports.

This paper has many similarities to the discipline of Agent Based Computation Economics, defined as "the computational study of economies modeled as evolving systems of autonomous interacting agents" (Tesfatsion, 2002). Agent based computation economics allows one to abandon the traditional role of "externally imposed coordination devices such as fixed decision rules, common knowledge assumptions, representative agents, and market equilibrium constraints." (Tesfatsion, 2002) and allow face-to-face interactions to be modelled in more complex ways.

Our objective is to describe a paradigm for thinking about integrating submodels based on the above perspective. The paradigm will suggest a software engineering framework, but it will also be shown that existing integrated models can be described according to the paradigm even before they are adapted into a new software framework.

1.2 Continuous time discrete event simulation

We are interested in the simplest representation of time that supports both fully dynamic models and a decoupling of the system for interactions from the details of the submodels. We represent time as a continuous variable, but everything will be modelled as if it occurred at instants. In reality the transactions between people often do occur at instants, but some things, such as physical processes, occur continuously. For things that, in reality, occur continuously, we offer two options:

- a "short term equilibrium" calculation, that reports the final result of a short term process at the instant when that process ends, or
- a numerical simulation, with time discretized to a degree appropriate to represent the process, and calculations reporting values at the instant of the end of each time slice.

We adopt the notion of a global time clock that schedules the various algorithms in each of the submodels. A global time counter moves forward from the simulation start to the simulation end, and at any instant in time the various submodels can be called upon to simulate events.

A common implementation of the global time clock is the "event queue". The event queue consists of a list of events that are scheduled to occur in the future. The list is sorted by the time at which the events are to occur. Each event has a "handler" (or "event routine", see Lin and Fishwick, 1996) associated with it, which is an identifier for a software module and necessary parameters and code entry points. The global time clock selects the event that has the lowest time, and executes its handler, causing code to execute in the appropriate software module to handle the event. The handling of one event often causes future events to be added to the event queue. The simulation ends when the event queue is empty, when the time clock reaches a predetermined stopping time, or when the user terminates the simulation interactively.

A discrete event simulation can be executed on distributed computing facilities in parallel, by dividing submodels, agents or interactions across processes on multiple processors. This requires either a "lookahead" value describing how far in time one process is allowed to get ahead of another process, or a "rollback" method to backup a process that gets too far ahead of other processes. (Lin and Fishwich, 1996).

2. Simulation structure

An interaction between two agents or two submodels is assumed to occur in two stages, the offer stage, and the transaction stage. The offer stage consists of an agent or submodel describing its ability or willingness to interact with other agents. The transaction stage consists of one agent accepting the offer of another agent.

2.1 The offer

An offer contains a description of the bundle that would be exchanged in a transaction. The bundle is described with a set of attributes:

- Category of offer (type of good, service or exchange)
- Direction (whether offering to supply/sell or demand/buy)
- Offerer

- Location of offer (x-y coordinate or polygon)
- Prices (for priced transactions in categories without "auctioneers")
- Attributes describing details of goods and services offered, and price and quality (exact list of attributes depends on Category)

2.2 The Auctioneer

In almost all real cases, sellers or buyers set their prices when offers are made. The description of the "offer" above respects this by including price as an attribute.

Yet most representations of economic markets rely on a story involving an "auctioneer" who controls the market itself, by adjusting the price to match supply and demand. The bulk of economic literature assumes a single price per "commodity" in each "place". This is a convenient abstraction, because it allows modelling of decision making agents as price takers, who do not adjust prices but only consider them. To allow for this convenience we provide, as part of the integration framework, an "auctioneer" algorithm which can be used for certain categories of goods and services.

The imaginary auctioneer adjusts an average price at a place and time. The price is fixed at that place and time until it is later updated by the auctioneer. Since it is an average price, the area of the place and the duration of the time should be large enough so that more than one transaction occurs in the category type at the price. Thus the use of the auctioneer forces aggregations of time and space and compromises our continuous space-time representation.

The auctioneer implemented within the integration framework is a simple non *tattonement* auctioneer, that adjusts prices over time in response to excess demand. There is no equilibrium assumption, and no "calling out" of prices asking agents what they would do, hypothetically, if prices were set at a certain point.

Representing a *tattonement* auctioneer, who somehow determines a market clearing price before any transactions are allowed, normally requires solving for equilibrium. This can be implemented into the framework, as long as the equilibrium is a short-run equilibrium that occurs over a defined period of time. During each such period the auctioneer must find the set of prices at the places so that supply and demand are equal. This is a tight coupling of supply, demand, and market mechanism, and a single algorithm in a single computational submodel solves for the equilibrium prices. The dynamic integration framework cannot (and should not) handle short term equilibrium calculations, so such equilibriums are relegated to the detailed and carefully crafted algorithms within submodels.

2.3 Offer list

"Offers" are statements of a willingness or ability to interact under certain conditions. We assume that most offers are *public* (meaning viewable, although not necessarily acceptable, by all other agents) and so we store all offers in a database of offers called the "offer list".

Integrating submodels into this framework then requires deciding for each submodel:

- what interactions it undertakes
- whether it does so as a supplier or demander or both,
- whether it makes offers or accepts offers, or both,
- when it makes offers or accepts offers,
- what conditions and prices it attaches to its offers, and
- what information regarding the history of the simulation does it need to decide what conditions to attach to its offers – e.g. past prices

2.4 Events for interactions

Since we have defined the simulation as a series of events in time, we are interested in the events that are required to allow interactions between submodels or agents. These can be classified into six standard event types and an additional non standardized event type:

- offer event
- unoffer event
- transaction event
- price-update event
- query offers event
- query completed transactions event
- short term equilibrium event (non-standardized)

Offer event: An offer event occurs when a submodel or agent publishes offers in the offer list. These can represent actual decisions that are made by agents at specific times to participate as an offerer, or a "block offer", which is a wholescale specification of available prices and/or quantities by a submodel that represents an aggregation of agents. Offers can contain a pointer to an acceptance handling routine (a "callback"), so that the offerer will be notified when its offer is accepted.

Unoffer event: An unoffer event occurs when an offer that was previously made (but not yet accepted) is withdrawn.

Transaction event: A transaction event occurs when a submodel or agent accepts an offer that is listed. For example, when a simulated individual goes shopping, she will be accepting a number of offers to sell goods and services. When a transaction event occurs the offer accepter takes some action. As well, the offerer may also log the acceptance and take some further action through the callback acceptance handling routine.

Price-update event: A price update event is associated with the auctioneer in a market for goods and services. The quantities purchased and sold in a geographic area since the last price-update event are compared to calculate the "excess demand". The change in the price is a function of the excess demand, tending to decrease prices when excess demand is too low, and increase prices when excess demand is too high. A price update event corresponds to an auctioneer adjusting the price for all current and future offers in a submarket. For homogenous goods and services, a price update event is often followed by the auctioneer offering infinite quantities of future demand and/or supply for the period of time until the next price update event. Auctioneers could also take into account outstanding offers and region wide excess demand in their price update algorithm, but the primary variable influencing the rate of price change is the local excess demand.

Query Offers event: This event causes a query to occur on the list of outstanding offers, finding the set of offers (or a random subset of the set of offers) that meet certain criteria. This is used by agents or submodels to determine a set of offers that could be accepted. This event does not affect the offer list directly, but in many cases a Query Offers event is followed by a Transaction event, where one of the offers returned by the query is accepted.

Query Completed Transactions event: Agents, submodels and auctioneers sometimes need to be able to observe and respond to market conditions. This method allows agents to inspect past history, to determine past averages or trends and use those averages or trends in their decisions to make offers or accept offers.

In addition to the standard six types of events, there may also be **short term equilibrium events** for some categories**.** A short term equilibrium event is associated with an imaginary auctioneer, but is also associated with agents – both suppliers and demanders. It represents a process whereby a price is set that causes the supply and demand in a submarket to be equal over a time period, and is usually associated with just one computational submodel. The equilibrium is necessarily a *short term* equilibrium, representing only a subset of the decisions that influence supply and demand, otherwise the equilibrium event would be a complete solution to the entire model system, and our basic assumption regarding the necessity of a time series simulation would be violated and this paper would not be applicable. The method for calculating a short term equilibrium set of prices is usually highly specialized, and so cannot be standardized in the same way the other six types of events are standardized.

All interactions between agents in the dynamic simulation occur through one of these event types. Thus the basic implementation of the interaction paradigm integration framework involves an implementation of the first six events types, plus special code to implement any short term equilibriums that are represented.

2.5 Derivative (abstract) categories – the Aggregator

Integrated policy analysis models are often concerned with small details in space and time (e.g. the success of local corner stores, intra-day traffic patterns) yet still hope to forecast the long-range future of a region. To preserve the full integration of decisions, the multitude of minor details are often "aggregated up" into a higher level, more abstract, statement of the availability of options, which influences the more major decisions. For instance, the attributes of travel between two points by the various available modes at the various times-of-day can be aggregated into a single composite utility number representing the difficulty of travelling between the two points. A common procedure in transportation modelling for aggregating detailed utility data is to use the logit model in the algorithm for choosing between details, which then allows using the "log-sum" of the available alternatives as the composite utility.

This can fit into the interaction paradigm if one imagines an aggregator who combines the options in the detailed offer lists, and offers an aggregate alternative in a more abstract offer list. This "aggregator" is not usually a real actor, but is no more abstract than the "auctioneer" of the typical story of how markets work. It is simply a description of how detailed information can be summarized for the benefit of submodels or agents.

3. Lag representations

3.1 Direct behavioural representation of lags

True lags occur in a system because of the rate of response of individual actors. If actors respond more slowly to changes in their situation, then there will be more lag. Actors respond slowly for two main reasons:

- it takes time and resources to notice and evaluate new conditions, and

- agents may understand that there is noise in the system, and forecast the future conditions based on some average of past conditions.

In a true dynamic microsimulation, the representation of lags would involve a direct behavioural representation of these two effects.

3.2 Abstract representation of lags – auctioneer stories

The direct behavioural representation of lags is an attractive modelling approach. Unfortunately it cannot be used for all types of submodels. When there is an "auctioneer" story regarding price setting we cannot use behavioural observation of the actual (non-existent) auctioneer to determine the speed of price response, but we can still ascribe certain time lags in the behaviour to the auctioneer.

There are two basic lag representations, representing the two auctioneer "stories". If the auctioneer story is setting a price to achieve a short-term equilibrium over a time step, then a submodel considers the elements of supply, demand and auctioneer simultaneously. The amount of lag can be controlled by adjusting 1) the length of the time step, and 2) what elements of demand and supply are included in the "short term" equilibrium. If the auctioneer story involves the auctioneer setting prices in response to calculated excess demand, then the amount of lag can be controlled by adjusting the amount of time over which the excess demand is calculated, and the rate at which the price changes in response to excess demand.

3.3 Abstract representation of lags – leapfrog equilibrium

If more than one type of commodity is subject to a short term equilibrium calculation, then it would be ideal if the short term equilibrium for both types of commodities could be calculated simultaneously. In practice separate submodels usually handle the two equilibriums, and each usually assumes that the prices established by the other are fixed. Thus there is a "leapfrog equilibrium" lag introduced by the joint response of each equilibrium calculation to the prices established by the other equilibrium calculation. This sort of lag is not directly behavioural, but could be interpreted as a delay in the flow of information regarding prices in one market to the decision makers in another market.

4. Example system – heterogeneous households and business establishments

A demonstration of the interaction of households and business establishments has been developed, to demonstrate the full heterogeneity possibilities of the system, and to begin to develop the software architecture implied by the system. The software implements the mathematics described by Abraham and Hunt (2001).

There are three types of "Economic Units" in the system: employed households, unemployed households, and business establishments. There are three categories of transaction: final goods (demanded by households and produced by establishments), intermediate goods (demanded and produced by establishments) and labour (demanded by establishments and produced by employed households).

The attributes of the economic units are the X-Y position (the residential location or business location), and a list of "expected parcels" which describe the wants and needs of the economic unit.

The initial population of economic units are synthesized by sampling each attribute from a specified distribution. Thus a group of economic units is described by a range of possible locations (a disc or a rectangle in a plane), a distribution for the *number* of expected parcels of different types to be associated with each economic unit, and distributions for the *attributes* of the expected parcels to be created.

Initially, the only events queued up in the event queue are the economic unit creation events, to synthesize individual random agents in space by selecting attributes from distribution. As each economic unit is created during the execution of these events the future events corresponding to the consideration of each if its expected parcels are added to the event queue. As each of the expected parcels are considered, the time to the next consideration of the expected parcel is sampled from a lognormal distribution and the next consideration of that expected parcel is added to the event queue.

For each event corresponding to the consideration of an expected parcel the economic unit searches the appropriate offer list for a set of candidate parcels, selecting random offers from three sets:

- The set of offers in the offer list that were offered by the economic unit who was the last successful trading partner for the expected parcel (if any).

- The set of offers with locations within a fixed radial distance from the economic unit's location
- The full set of offers of the appropriate direction in the offer list

A logit model is then used to select from the candidate parcels, or from the additional alternative of making an offer. If the decision is made to make an offer (rather than to accept an offer) then the offer is made with attribute values identical to the expected parcel.

A nested structure is used, with two random components in the utility function, one associated with the top level of whether to make or an accept an offer, and one associated with the bottom level of which offer to accept conditional upon deciding to accept an offer. Thus the attractiveness of each considered offer is

$$U_{co} = V_{co} + \varepsilon_{co} + \varepsilon_{accept}$$

where ε_{co} is the random component associated with the individual offer under consideration,

ε_{accept} is the random component associated with the choice to accept one of the Considered Offers, and

V_{co} is the non-random component of the utility of the Considered Offer, calculated as described below.

The attractiveness of making an offer is

$$U_{mo} = V_{mo} + \varepsilon_{mo}$$

where ε_{mo} is the random component of the utility of making an offer, which has the same distribution as ε_{accept}

V_{mo} is the non random component of the utility of making an offer, described below.

Note that there is no need to calculate the "log sum" term for the expected maximum value of selecting from the set of considered offers. We are sampling the error terms directly from the underlying distributions, so we know the actual maximum utility, and have no interest in the expected maximum.

The "non-random" component compares the set of integer and real attributes used to describe both the expected parcel and the offered parcel. It is called "non-random" because the random number generator does not need to be invoked to calculate this component of utility, but it is important to note that it compares expected parcels that were initially generated using the random number generator.

Transport is considered in the model, but only as a derived demand. Thus transport is not in the list of expected parcels. Instead, the transportation system is assumed to offer infinite travel at a set of fixed costs and times. When an expected parcel is under consideration, the transportation necessary to travel from the economic unit's location to each offer's location is considered as well.

Thus, the utility of any offer is given by

$$V_{co} = \sum_i f_i\left(X_i^{CO}, X_i^{EP}\right) - TC_{L_{CO}, L_{EP}}$$

where X_i^{CO} is the value for attribute i for the Considered Offer,
X_i^{EP} is the value for attribute i for the Expected Parcel, and
$f_i()$ is a functional form for evaluating the attribute i of the Considered Offer and comparing it, if necessary, to the attribute of the Expected Parcel.
$TC_{X,Y}$ is a measure of the disutility of travelling between location X and location Y
L_{CO} is the location of the Considered Offer
L_{EP} is the location of the Expected Parcel, which is the same location as the Economic Unit.

The non-random component of the utility of publishing an offer rather than accepting an offer is based on the attributes of the Expected Offer, plus a constant that is fixed for each class of commodity for either acquiring (e.g. buying) or divesting (e.g. selling):

$$V_{mo} = \sum_i g_i\left(X_i^{EP}\right) + K_{CC}^{D}$$

where $g_i()$ is a functional form for evaluating the attribute i of the Expected Parcel.
K_{CC}^{D} is the constant associated with commodity class CC and direction D
D is either "acquiring" or "divesting".

The functions $f_i()$ and $g_i()$ control how the attribute i influence the choice. In the demostration there are three types of attributes. One type, called *unidirectional,* measures quality or quantity and always influences the utility in the same direction, so:

$$f_i\left(X_1, X_2\right) = g_i\left(X_1\right) = \theta_i \cdot X_1$$

where θ is a parameter controlling how increasing values of the attribute increases the value of the utility.

The second type, called *bi-directional*, measures the difference between what is expected and what can be obtained through accepting an offer, with a negative parameter so that a better match improves utility:

$$f_i\left(X_1, X_2\right) = \theta_i \cdot \left(X_1 - X_2\right)^2$$
$$g_i\left(X_1\right) = 0$$

The third type, called *equals only*, measures whether the offer under consideration matches exactly in the attribute:

$$f_i(X_1, X_2) = \begin{cases} \theta_i for X_1 = X_2 \\ 0 otherwise \end{cases}$$
$$g_i(X_1) = \theta_i$$

To achieve a dynamic trend towards equilibrium the Expected Parcels are updated based on the ability to enter into a transaction to acquire or divest a parcel similar to the expectations. If, over time, an Economic Unit continually chooses to accept an offer rather than make an offer when considering an Expected Parcel, then some attributes of the Expected Parcel will be adjusted to be more optimistic (price would adjusted down for a buyer of a commodity, and adjusted up for a seller.) Similarly, if an Economic Unit decides to make an offer rather than accept an offer, and the Economic Unit's offer is quickly accepted by another Economic Unit, then the Expected Parcel will be adjusted to be more optimistic. On the other hand, if an offer is made and the offer is not accepted, or is only accepted after some delay, then the Economic Unit will adjust its Expected Parcel to be more pessimistic.

The attributes of the expected parcels, the distribution used to generate them for the economic unit, the form of their influence on the utility, and whether they are adjusted in the long term response, are shown in Table 1.

This simulation system has been implemented in software, and is currently being tested to show that it can reach a stable state, and that the stable state will be sensitive to changes in inputs according to microeconomic theories regarding supply and demand and spatial price equilibrium.

Currently there is no "location choice" occurring in this simulation. Location, like transport, is a derived demand in that a good location allows an economic to meet its wants and needs with less travel. A strategy for considering location within this type of simulation is currently being developed. It is loosely based on genetic algorithms.

Table 1 Attributes of Expected Parcels in Establishment-Household Demonstration

Economic Unit	Expected Parcel	Attribute	Distribution	Influence on Utility	Expect. Adjust
Households as buyers, Establishments as sellers	Retail goods	Quality	Log Normal	Unidirectional	
		Price	Uniform	Unidirectional	Yes
		Detailed Type	Uniform	Equals only	
		Quantity	Uniform	Unidirectional	Yes
Employed households as sellers, Establishments as buyers	Labour	Hourly wage	Log normal	Unidirectional	Yes
		Shift preference	Uniform	Unidirectional	
		Hours weekly	Lognormal	Bi directional	Yes
		Occupation	Uniform	Equals only	
		Generic / unknown 1	Normal	Bi directional	
		Generic / unknown 2	Normal	Bi directional	
Establishments as buyers and sellers	Inter-mediate goods	Quality	Log Normal	Unidirectional	
		Unit Price	Uniform	UniDirectional	Yes
		Detailed Type	Uniform	Equals Only	
		Quantity	Uniform	Unidirectional	Yes
		Generic / Unknown	Normal	Bidirectional	

5. Example System – Oregon statewide model.

The Oregon Statewide model is being developed by the Oregon Department of Transportation for policy analysis. It is described in Hunt *et al*, 2001. This model has been developed without the benefit of the framework, but we show how the framework could be used to standardize the interactions between the submodels. We describe the model in a way that is true to the theoretical design and mathematical implementation of the actual model, but as if the software implementation was consistent with the interaction framework. The submodels and interactions of the Oregon Statewide model are shown in Figure 2.

5.1 Submodels

The model consists of seven modules:

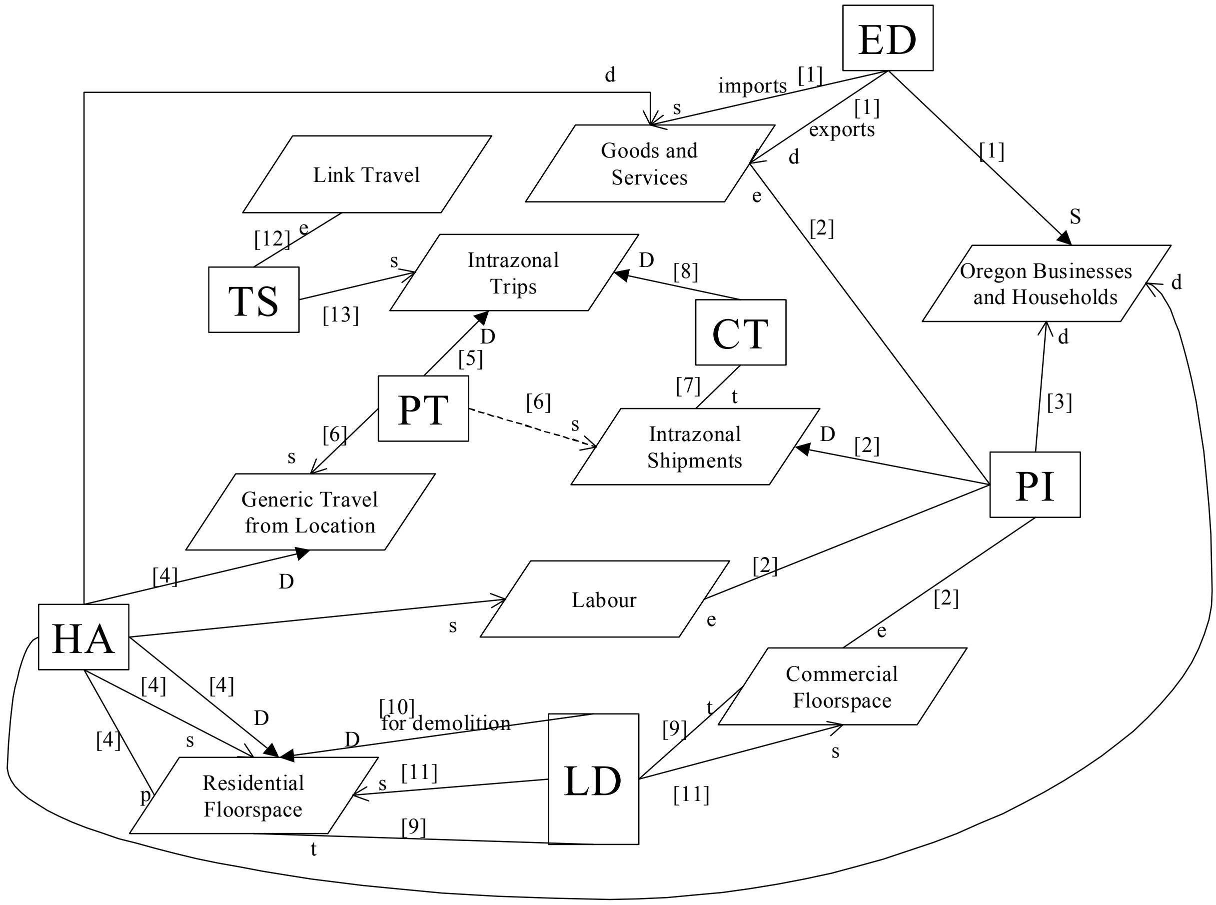
ED: Regional Economics and Demographics The ED module provides the rest of the model with regional control total for production by economic sector, inputs and exports by economic sector, employment by labour category, in migration, and payroll by sector for each year.

PI: Production Allocations and Interactions The PI module determines for each year the distribution of production activity (industry and employment) among zones, the consumption of space by production activity, the flows of goods and services and labour from the location of production to the location of consumption, and the short term equilibrium prices for goods, services, labour and space. An aggregate allocation framework is used which allocates total production to zones, allocates technology within an industry in a zone, and allocates inputs and outputs in space. The system solves for the set of prices in space which clear each local (zonal) submarket for each category of good, service, labour and floorspace. The PI module is an implementation of the Activity Allocation submodel of PECAS (Hunt and Abraham, 2003), but with residential location choice by households removed.

HA: Household Allocation The HA module uses a microsimulation of the behaviour of individual households to determine changes in household composition, household actions regarding home location (and residential space use), final demand by households, household car ownership, employment status, employment location and career choice, and school status, school location and education choice. The price of residential floorspace is updated in HA based on the amount of vacancy.

LD: Land Development The LD module determines the actions of land owners and developers regarding the quantity and type of buildings. It uses a fine grid representation of the land and the associated regulations and characteristics regarding developability, and responds to the prices that are established for floorspace.

CT: Commercial Transport The CT module determines truck movements arising during a representative workday in the year. It takes the flow quantities by commodity type established in PI, and assigns them to modes and vehicles. Thus it converts economic flows to vehicle movements.



s = offer to sell/supply (supply)
d = offer to buy/demand (demand)
S = sell/supply (accept offer to buy) - also implies query of offers
D = buy/demand (accept offer to sell) - also implies query of offers
q = query the list of offers (without immediately accepting some)
t = query the list of completed transactions
p = price update
e = short term equilibrium event
accept (transaction event, S or D)
offer (offer event, s or d) with callback
offer (offer event, s or d) without callback
[1] sequence of events

Figure 1 : Markets and modules in the Oregon Generation 2 Model

PT: Personal Travel The PT module establishes a list of the specific individual trips made by members of households during a representative day. It establishes for each trip a starting location, ending location, starting time, tour mode (mode choice) and vehicle occupancy. The process establishes an activity pattern for each person for the day (which is a listing of the sequence of activities undertaken by the household member), and then establishes the time of departure for each trip and the location for each stop on the trip.

TS: Transport Supply The TS module assigns the individual trips established in PT and CT to a network of transportation links. A shortest path algorithm is established for each trip, and after each trip's path is determined the performance of the links used by that trip are updated to account for congestion. Each tripmaker's preferences regarding route attributes are randomly selected. The algorithm iterates to a converged state, so that individual trip makers are choosing optimal routes given their preferences. The TS module reports link performance and zone-to-zone travel times and costs for a representative traveller.

5.2 Categories of interaction

The connections between these submodels can be viewed in the offer-accept interaction paradigm:

The **Link Travel** category: This is the demand and supply of the physical space on a transportation link. In the short term the capacity is fixed. The Transport Supply module adjusts the routes of the trips in its trip list so that each traveller is choosing an optimal (for them) route, given the congested travel conditions. This is a short-term equilibrium event, and the TS module represents the behaviour of the supply, the demand, and the imaginary auctioneer process for establishing the set of link demands and the price-like signals of the travel time for each link.

The trip list reflects the short term demand, the fixed network reflects the short term supply. The long term supply is adjusted by the model user; the trip list comes from the Interzonal Vehicle Trip market.

The **Interzonal Vehicle Trip** category: This is the market for vehicle trips between pairs of locations. The Transport Supply module determines the travel distance and time by different modes and offers them as a set of matrices of representative times and costs for origin-destination pairs, using the zone system to delineate the origins and destinations. Thus TS offers an infinite quantity of travel between zone pairs, and publishes the money cost and travel times for a typical traveller for a particular mode at a particular time of day. Two other modules accept these offers: Personal Travel and Commercial Transport. PT and CT decide on the choices of where and when to travel for persons or shipments. TS represents the supply, and PT and CT represent different facets of demand.

The **Intrazonal Shipments/Flows** category: This is a derivative category, an aggregation and summary of the Intrazonal Vehicle Trip market. The Intrazonal Vehicle Trip category is for particular modes and at particular times of day. The Production Allocations and Interactions (PI) module does not work in terms of specific times of day or modes, but instead requires composite measures of the deterrence function for shipping between origin and destination. The PT module acts as an aggregator, calculating the expected maximum value of the mode and time-of-day decision, and offers the "log sum" value. The log-sum is a price-like signal, and it is offered in infinite quantity for a one year period. PI accepts those offers, leading to a matrix of interzonal flows that have not been assigned a mode or time-of-day.

The **Generic Travel From Location** category: This category is an even more abstract view of travel. It represents the type of travel that households of a particular category make given a particular home location. The Personal Travel module PT again acts as an aggregator, calculating an accessibility value for different household categories as a price-like signal for household travel. These are offered in infinite quantity for a 1 year period. The HA module accepts those offers when it locates a particular household in a particular location.

The **Residential Floorspace** category: The price of residential floorspace in each zone is established in HA through a price update function, which compares the vacancy rate in a zone with an equilibrium vacancy rate, and adjusts the price of all offers-to-let upwards if the vacancy rate is too low and downwards if it is too high. This allows a finer time-resolution in the response of prices – currently prices are updated monthly. All offers in this market are thus "market price" offers, in that there are no prices set by the offerers. The Land Development module offers newly constructed floorspace for rent. This new construction comes on once per year when the LD module is run. LD simulates developers decisions, and developers in this model make decisions based on the set of current prices, effectively assuming that future prices are likely to equal current prices. LD measures current prices by querying the transactions that have occurred (the 't' event in Figure 1). The Household Allocation module accepts offers of floorspace when a household chooses a new location. If the household is vacating an old location, that location is offered to the market, and becomes vacant. To accomodate demolition and reconstruction of occupied properties there is a continuous offer by each tenant to give up their property and supply it for demolition, and these offers are accepted by land-owners when they begin demolition.

The **Labour** category: The labour market is segregated by zone. The Production Allocation and Interactions module (PI) establishes a short term equilibrium of demand and supply of labour in each workplace location, and establishes a price for that labor. The demand for labour is calculated within PI directly. Since this is a short-term equilibrium calculation, PI must also have a representation of the supply of labour in each zone. PI has it's own simplified model of workplace destination choice for households to establish the supply schedule in each zone.

The **Oregon Businesses and Households** category: PI and HA report on overall economic conditions in Oregon, and provide that input to ED for it's decisions regarding in migration. This can be put into the market paradigm by saying that PI and HA make fixed-price (actually, fixed utility) infinite quantity offers to move new households and new business activity into the model region. ED will accept a certain quantity of those offers.

The **Commercial Floorspace** markets: LD offers a fixed quantity of commercial floorspace in each time period. PI establishes a short term equilibrium with demand, accepting the offers of commercial floorspace, and establishing the price in each zone. These transaction prices are queried by LD to aid in developer decision making, with developers assuming that future prices will be the same as past year prices.

The **Goods and Services** markets: HA establishes the location and number of households, and hence establishes the final demand for goods and services by zone. These are taken as offers to purchase goods and services. ED establishes the import and export of goods and services from the model region: these are taken as offers to purchase and/or sell goods and services by agents exogenous to the simulation. These lists of offers are then all accepted by PI when it establishes the spatial distribution of goods flows. Intermediate flows of goods and services between businesses in Oregon are established within this PI short term equilibrium.

5.3 Timing

The model runs in one year time steps, in sequence through ED, PI, HA, CT, PT, LD and TS.

ED accepts offers to migrate businesses and households into the model region, and updates the offers to purchase and sell goods by importers and exporters.

PI then solves the short term equilibrium of business location, goods and services movements in abstract, labour, and commercial floorspace prices. In doing so it accepts household's offers to travel to work to supply labour and to purchase retail goods and services, landowners offers of commercial floorspace for rent, the offers to purchase and sell goods and services by importers and exporters, and the intrazonal shipment offers. PI then updates its own offer in the Oregon Business and Households market to accept new businesses in each industry category.

HA then moves households around, with households accepting offers for residential space, and adding new offers to let residential space corresponding to space vacated during a move. HA considers the accessibility numbers corresponding to offers of "generic travel given location" when choosing a location for each household. HA assigns particular jobs to individuals in households based on the flows established in PI. During the course of the HA operation, the prices of residential floorspace are updated by 12 price-update events. Thus HA is viewed as taking the full 12 months to complete, with a price update at the end of each month.

At the end of the year PT executes, generating the travel associated with households and businesses, and accepting offers in the Intrazonal Vehicle Trip market. PT updates the offered prices in the Intrazonal Shipments market, and in the Generic Travel Given Location market.

CT then executes, taking the intrazonal shipments that were accepted by PI and converting them to intrazonal vehicle trips, and accepting intrazonal vehicle trip offers. LD then executes, querying the residential floorspace prices established by HA and the commercial floorspace prices established by PI, and simulating the actions of developers who assume those prices are a guide to future rent revenue.

LD updates the offers to lease space in the commercial floorspace offer list and the residential floorspace offer list. When some residential floorspace is demolished or redeveloped, LD takes up households' offers to vacate their premises, evicting them.

Finally, TS takes all of its offers of intrazonal vehicle travel that were accepted, and converts them into link travel by solving a short term equilibrium of congested assignment. TS then updates its intrazonal travel offers.

5.4 Potential changes

The Oregon model contains some good example of how the dynamic transaction simulation paradigm could be more strongly adapted to further decouple modules and improve performance.

TS, PT and HA each process items individually: TS processes trips, adding them and removing them from links. PT processes individuals, choosing trips for them. HA processes households. TS offers intrazonal travel attributes to PT, and PT offers accessibility to HA. These could be run in parallel, with:

- PT processing individuals in a household once HA signals the end of its processing of a household by accepting an accessibility offer published by PT, and
- TS processing each trip for an individual as PT accepts TS's offers of intrazonal vehicle trips.

PI currently searches for a set of prices for each good or service, and each commercial floorspace type, in each zone, to clear each such market. PI could be made dynamic by replacing these price search processes with price update processes, with the same PI equations used to allocate consumption and production of goods and services and business location given a set of prices, but the set of prices change over time in response to shortages and surpluses, rather than to an equilibrium state in each time period that clears the market. This would make PI dynamic and would decrease computational burden, as the PI equations would not have to be repeatedly executed to establish an equilibrium (see also Abraham and Hunt, 2002).

6. Conclusions

This work developed a paradigm for *integrating* model components and agent representations of behaviour. Since integration involves interactions, since interactions involve some exchange, and since markets are sometimes defined as mechanisms facilitating exchange (Katzner, 1988), this conveniently allows modelling to follow economic theory.

There is a built in system (using the *non-tatonnement* auctioneer) for using prices (or price-like variables) to match the quantity of demand to the quantity of supply and avoid (ongoing) shortages or surpluses. But a behavioural process for matching prices would involve agents adjusting their expectations and offer prices over time based on their own personal past history and the history of similar transactions.

The representation is shown to be relatively simple, involving only six regular operations: make offer, withdraw offer, accept offer, query offers, query completed transactions and update market price. These six operations allow a decoupling of supply and demand elements in each market, and hence allow a decoupling of the various parts of a simulation and the various agents that interact to evolve cities and regions. Decoupling is desireable to facilitate plug-in code modularity and expansion, separate submodel development teams of analysts and programmers, and facilitate parallel execution on distributed computing facilities.

A seventh type of operation, a short term equilibrium event, requires a tighter coupling of supply, demand and market into a single equilibrium search algorithm. The concept of equilibrium does have certain advantages, and is so engrained into practice and economic research that supporting it seems essential. This paper shows how short term equilibrium events in some markets can be coupled with dynamic disequilibrium market representations for longer term trends in the same markets, and with fully dynamic representations in other markets. However this equilibrium event is a bit of a "force fit" into the paradigm – the true power of the paradigm emerges when equilibrium representations are replaced by fully dynamic representations.

A system of heterogeneous agents, continuous time, and continuous space was described, hinting how a full agent based microsimulation can take full advantage of the paradigm. In addition, an existing modelling system was investigated, to show how it can fit into the paradigm conceptually. Each of the individual submodels in this system could be wrapped up in interface code to fit into a software implementation of the paradigm.

The paradigm has a continuous time representation as its basis, and supports a continuous space representation. Existing modelling systems usually work in terms of time steps and aggregate space. It is disappointing to have to forgo the full potential of the market paradigm in this regard when taking on existing algorithms and submodels, but the alternative of throwing out most existing work and rebuilding from scratch requires too much of a risky commitment of research time and money to be appropriate for any planning agency, or even most research institutions. The work described here provides a forward path that may eventually lead to fully dynamic and disaggregate agent based continuous time microsimulations, but that also has many safe and useful stops in the near future.

ACKNOWLEDGEMENTS

This work benefited from conversations with Eric Miller and Paul Salvini. The work was supported by the Natural Sciences and Engineering Research Council of Canada and the Social Science and Humanities Research Council of Canada through a Major Collaborative Research Initiative grant.

7. REFERENCES

Abraham, J.E. and J.D. Hunt, 2001, "Dynamic Microsimulation of Heterogeneous Spatial Markets", presented at the Workshop on Economics with Heterogeneous Interacting Agents, University of Maastricht, The Netherlands, June 7-9 2001

Abraham, J.E. and J.D. Hunt, 2002, "Spatial market representations: concepts and application to integrated planning models", presented at the *49th annual North American Meetings of the Regional Science Association International* in San Juan, Puerto Rico, November 14-16, 2002

Hunt, J.D., R Donnelly, JE Abraham, C Batten, J Freedman, J Hicks, PJ Costinett and WJ Upton, 2001, Design of a Statewide Land Use Transport Interaction Model for Oregon, World Conference on Transportation Research, Seoul, South Korea

Hunt, J.D and J.E. Abraham, 2003, Design and Application of the PECAS Land Use Modelling System, proceedings of the 8th International Conference on Computers in Urban Planning and Urban Management (CUPUM), Sendai, Japan, May 27-29, 2003

Katzner, D.W., 1988, *Walrasian Microeconomics*, Addison-Wesley, Reading, MA

Lin, Y and PA Fishwick, 1996, Asynchronous Parallel Discrete Event Simulation, *IEEE Transactions on Systems, Man and Cybernetics,* Vol 26 No. 4 pp 397-412

Tesfatsion, L, 2002, "Agent-Based Computational Economics", Iowa State University Economics Working Paper No. 1, Revised July 2002, accessed at http://www.econ.iastate.edu/tesfatsi/acewp1.pdf on Dec. 19, 2002.