Project no.
035086

Project acronym
EURACE

Project title
An Agent-Based software platform for European economic policy design with heterogeneous interacting agents: new insights from a bottom up approach to economic modelling and simulation

Instrument STREP

Thematic Priority IST FET PROACTIVE INITIATIVE "SIMULATING EMERGENT PROPERTIES IN COMPLEX SYSTEMS"

Deliverable reference number and title
D2.1 Agent-based computational economic modelling guidelines

Due date of deliverable:
31-08-2007

Actual submission date:
09-10-2007

Start date of project: September 1st 2006 Duration: 36 months

Organisation name of lead contractor for this deliverable
Université de la Méditerranée - GREQAM

Revision 1

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Executive summary

The deliverable for WP2 (D2.1) consists of two documents that contain information on the current state-of-the art in agent-based computational modelling in economics.

The model development requirements for the framework are specified in the document entitled 'Modelling Requirements for EURACE'. A general description of the current status of our modelling effort is provided in 'Modelling Specifications for EURACE' that gives further details on the model components.

General aim The aim of these documents is to construct canonical or generic models of economic interaction on the goods, labor and asset markets with the objective to arrive at a unified modelling framework that can be used in all the Work Packages within the EURACE project.

The second aim of the documents is to provide a broad catalogue of the modelling options. For this reason it has been written as a broadly scoped review of market mechanisms. It is most likely that not all processes and ideas stated herein will finally end up in the EURACE simulator. But nonetheless, it is our hope that it provides a sufficient number of options to make a well-informed choice out of the myriad of modelling possibilities.

Modelling methodology A general methodological problem for agent-based models is the appropriate design or selection of decision rules that govern the behavior of individual agents. We discuss the following items at a general level of detail:

- Agent types: their allowable actions and methods, their behavioral repertoires.
- Market types: their mechanisms and which agents can be active on which market.
- Environment types: the institutions and restrictions that limit the activities of the agents.
- Interaction structures: agent-agent, agent-environment, environment-environment interactions.
- Simulation runs: output, storage and graphical representations.

One of the most important tasks in agent-based modelling consists of specifying the interaction structure. It is also one of its greatest challenges. Issues of local versus global interaction, the handling of time, the sequencing of events, and how we deal with economic processes that take time; all these aspects impose a great responsibility on the modeller to specify completely and unambiguously the economic environment and how the agents interact within that environment.

As far as firm behavior is concerned, standard decision rules and heuristics have been developed for many operational decisions. These rules are well documented in the relevant business and operations management literature (e.g., for decisions related to pricing, production and inventory management, and market entry decisions). Our general 'philosophy' in terms of modelling the firm's behavior is to implement relatively simple decision rules that match as much as possible the standard procedures of real-world firms, as described in the corresponding management literature.

Similarly, the decisions of consumers are modelled according to simple empirically founded rules which can be obtained from the literature on consumer behavior (e.g., decisions relating to the allocation of disposable income to consumption or savings, portfolio selection decisions on the asset market, or the choice of firm outlets for shopping decisions).

Modelling guidelines The 'Modelling Specifications for EURACE' provides information on the geographical scope of the models, a broad categorization of the economic sectors and agent classes, and the modelling guidelines for each market (consumption goods, investment goods, labour, credit, and financial assets).

The purpose of the modelling guidelines is to provide high-resolution descriptions of the modules to be used in the EURACE simulator. These descriptions should be at such a high level of detail that it yields sufficient information for the computer engineer to implement the model element. For example, for the agent types the model specification will not only list all allowable actions for a particular agent type, but it will also need to specify the processes underlying these actions, and under what conditions a particular method is being activated.

Modelling Requirements for EURACE

Responsible authors:
Sander van der Hoog
Christophe Deissenberg
(GREQAM, Université de la Méditerranée)

Contributions from:
Andrea Teglio (Genoa)

Workpackage:	WP2.1 : Agent-based computational economics
Date:	October 8, 2007
Contributing units:	Genoa
Responsible unit:	GREQAM, Université de la Méditerranée
Distribution level:	Public domain
Discussed at meeting(s):	Ancona, 27 October 2006; Nice, 26-28 January 2007.

Contents

Abstract

This document presents the initial modelling requirements for the development of the EURACE simulator. It is a survey of market mechanisms, trading protocols and learning algorithms. Its main purpose is to provide a catalogue of modelling choices and to serve as a background survey for the EURACE project as a whole.

Acknowledgements

This work was supported by the European Union through its 6^{th} Framework Programme, FET-IST 'Complex Systems'. Funding for the STREP research project 'EURACE' under contract no. 035086, is gratefully acknowledged. The following persons have contributed to this paper. Andrea Teglio contributed to Section 7 on learning mechanisms. Discussions with Eric Guerci on the electricity markets in Section on 6.2 are gratefully acknowledged. General discussions on the topics in this paper have been held at various stages of development during the EURACE meetings. The first was held in Ancona, 27 October 2006 and the second was held in Nice, 26-28 January 2007. The usual disclaimer applies.

1 Introduction

An initial development of the modelling requirements has already taken place in the project proposal document, but requires further expansion. The model development cycle begins with a description of the general characteristics of the proposed model and a complete written description of all model components. These model development requirements are to be specified in a Modelling Requirements Document (MRD). After this very general description the modelling components need to be further detailed in a Modelling Specifications Document (MSD). For EURACE we propose the following working titles for these two key documents:

- MRD: 'Modelling Requirements for EURACE' (this document).
- MSD: 'Modelling Specifications for EURACE' (see van der Hoog and Deissenberg, 2007).

The MRD lists the following items at a very general level of detail:

- Agent types: their allowable actions and methods, their behavioral repertoires.
- Market types: their mechanisms and which agents can be active on which market.
- Environment types: the institutions and restrictions that limit the activities of the agents.
- Interaction structures: agent-agent, agent-environment, environment-environment interactions.
- Simulation runs: output, storage and graphical representations.

The MSD provides more in-depth descriptions for each item listed in the MRD. The descriptions should be at such a high level of detail that it yields sufficient information for the computer engineer to implement the model element. For example, for the agent types the MSD will not only list all allowable actions for a particular agent type, but it will also need to specify the processes underlying these actions, and under what conditions a particular method is being activated.
The high-resolution descriptions of the MSD should satisfy the criterion of Dynamical Completeness (see Tesfatsion, 2006). Ideally, the MSD should also contain for every model component a Level of Effort (LOE), providing an estimate of how long it will take to implement a particular model element.

The first criterion for the successful completion of WP2 is: "Acceptance of the guidelines by the project partners." (C2.1). The second criterion is: "Successful peer-reviewing of the guidelines by outside experts." (C2.2). The internal release of the final versions of the MRD and MSD to the project participants will constitute having satisfied criterion C2.1. Criterion C2.2 will not be met until the outside experts have given their approval. Only when both criteria C2.1 and C2.2 have been met, will this constitute the end stage of the "Description of the Modelling Guidelines" and therefore signal the conclusion of WP2.

2 Background

Pryor et al. (1998) propose the following procedure for building high-fidelity, high-resolution agent-based models:

> 'To build [such] a computer model [...] requires an initial in-depth research and writing project, which is the level of effort (LOE) [...]. Information must be gathered about all of the actors identified so that we understand how they behave [...]. [...] We must develop a set of operations that the actors perform and then define the applicable operations in a logical sequence. We must be able to identify and quantify the resources on-hand and remotely accessible to the actors.
>
> The result of this research and writing project would be a Requirements Specification Document. This document would provide a detailed description of how the computer model would be implemented, including finalized details about the scenarios, the actors, and the simulation output. The document would also contain a complete description of the operational behavior and attributes of each actor, including the actor's linkages with other actors.' (Pryor et al., 1998, p. 13)

> 'The individual system models (those constructed for each of the actors in the proposed computer model) would likely constitute a reusable library of agents. Written in an object-oriented language, these agents could be dragged and dropped into different scenarios. With little or no modification, the agents could be applied to other [scenarios].' (Pryor et al., 1998, p. 13)

3 Unified modelling framework

A general methodological problem for agent-based models — that attempt to avoid the overly strong assumptions of equilibrium analyses such as informational requirements and the rationality of individuals — is the appropriate design or selection of decision rules that govern the behavior of individual agents. Deviation from the intertemporal (constrained) maximization paradigm opens many degrees of freedom with respect to the type of behavioral rules that can be used and the way the behavior is adapted over time. However, as far as firm behavior is concerned, for many operational decisions standard decision rules and heuristics have been developed that are well documented in the relevant business and operations management literature (e.g., for decisions related to pricing, production and inventory management, and market entry decisions). Our general 'philosophy' in terms of modelling the firm's behavior is to implement relatively simple decision rules that match as much as possible the standard procedures of real-world firms, as described in the corresponding management literature. This modelling approach can be seen in the same spirit as the seminal work by Cyert and March in the 1960s (Cyert and March, 1963/92) on the behavioral theory of the firm. Similarly, the decisions of consumers are modelled according to simple empirically founded rules which can be obtained from the literature on consumer behavior (e.g., decisions relating to the allocation of disposable income to consumption or savings, portfolio selection decisions on the asset market, or the choice of firm outlets for shopping decisions).
To summarize, the modelling philosophy we adopt is to ground our behavioral models in empirical reality by using as much as possible the behavioral rules that are used in the real-world. This seems to give an adequate response to the argument that there are *'too many degrees of freedom in modelling bounded rationality'* (Sargent).

Aims of this document

The aim of this document is to construct some canonical or generic models of economic interaction on the goods, labor and asset markets with the objective to arrive at a unified modelling framework that can be used in all the Work Packages within the EURACE project. Such a unified modelling approach broadly consists of two parts:

- Defining the economic environment.
- Describing the interactions among the agents in the environment.

The economic environment needs to be fixed ex ante by the modeller, as are the behavioral repertoires of all the agents. While the aim of ACE is to model agents as completely autonomous entities and to let all agent interactions be determined endogenously, full agent autonomy (to decide exactly how and with whom to interact) will not be possible: the modeller always has to specify the behavioral repertoire of the agents in the model. An agent cannot act outside of this behavioral repertoire, but the final outcome of its behavior depends on its interactions with its environment and with the other agents.
Specifying the interaction structure is one of the most important aspects of agent-based modelling, but it is also one of its greatest challenges. Issues of local vs. global interaction, the handling of time, the sequencing of events, how we deal with economic processes that take time; all these aspects impose great responsibility on the modeller to specify completely and unambiguously the economic environment.
The second aim of this document is to provide a broad catalogue of modelling options. Therefore it has been set up as a broadly scoped review of market mechanisms. It is most likely that not all processes and ideas stated herein will finally end up in the EURACE simulator. But nonetheless, it is our hope that it provides a sufficient number of options to make a well-informed choice out of the myriad of modelling possibilities.

The next sections provide further information on the handling of time, interaction structures, market types, agent environments, learning algorithms, and the simulation output and data storage requirements.

4 The handling of time

[This section is based on a presentation on agent activation regimes by Robert Axtell (CEEL Summer School, Trento 2006), and the paper Axtell (2000).]

4.1 Agent activation regimes

The result of an agent-based simulation may be sensitive to the timing of the agents' activities (see, e.g. Axtell (2000)). Therefore it is important to check whether changes in the agent activation regime affects the outcome of a simulation. If a simulation model is not robust against such changes the simulation environment may produce software artifacts that are solely due to the fact that agents perform their activities in a certain order. The main choice in regime is between a parallel and serial activation:

- Parallel activation: agents are performing their activities in parallel, interacting synchronously or asynchronously depending on their interdependencies.

- Serial activation: the activities of agents occur in a predetermined sequence, which can be randomized according to some stochastic distribution (e.g., uniform or Poisson distributed).

4.1.1 Parallel activation

For parallel activation we can further differentiate several activation modes:

- Synchronous parallel activation: All agents move in lock-step, given the previous period's state information (e.g., the updating of cells in a cellular automaton occurs for all cells simultaneously).
- Partially asynchronous parallel activation: Agents act in parallel and communicate as possible (waiting delays are bounded). This means that agents sometimes may need to wait for the activity of another agent to finish before they can start with their own activity. The agents' activities are interdependent, but if the waiting time becomes too long the agent gets activated (this may be related to a 'time-budget').
- Fully asynchronous parallel activation: Agents act in parallel without any guarantee on delays and the subsequent inefficiencies this may produce for the system as a whole. This means the agent activation is fully interdependent and agents have to wait for as long as it takes for other activities to finish.

4.1.2 Serial activation

Serial activation means that every agent acts one at a time. Below we list several possibilities for serial activation regimes:

- Uniform Activation: In a Uniform Activation regime all agents are activated in every period, in a predetermined order. This implies that no agent is inactive in any period. This activation regime has as a disadvantage that the order of agent activation may cause artifacts. The solution would be to randomize the order of agent activation.
- Random Activation: In a Random Activation regime the agents are selected to be active with uniform probability. The advantage of this method is that it has a fast implementation. The disadvantage is that it is not behaviorally credible, since not every agent may be active with equal probability. Using this method, a period can be defined by A agents being activated, where A is a number smaller than the total agent population. This method thus has as a feature that a fixed number of agents is active in every period, but some agents may be more active than others. A disadvantage is that not all agents are active in every period.
- Poisson Clock Activation: Using a Poisson Clock Activation regime, each agent has an internal clock that wakes it up periodically according to a Poisson distribution. This method has the appealing feature that it is behaviorally more credible than the Random Activation regime since there is 'true' agent autonomy. Each agent wakes up at a random time to perform certain activities independently from any other agents. A 'period' can then be defined as the amount of 'wall time' that elapses until A agents have been activated on average. Another positive feature of this method is that there is a variable number of agents that are active in every period, and some agents are more active than others. A disadvantage of the Poisson Clock Activation regime is that agents

must be sorted every period, since they have an individual activation schedule stating at which time the agent has to perform some task. This individual activation schedule is obtained by sorting the random list of activation times, for each agent separately, and then constructing the global activation schedule for all agents. This requires a concatenation and sorting of all the individual activation schedules.

Implementation of Poisson Clock Activation regime

- Specify at the beginning that the model will be run for T periods.
- At time 0, for each agent $a \in A$ draw T random numbers as follows: $t_i + 1 = t_i - log(U[0,1])$.
- Sort these NT random numbers to develop the *activation schedule*: Naive sort scales like N^2, Quicksort scales like $Nlog(N)$.

With a population size of $N = 10^6$ this means that the sorting scales on the order of $10^6 log(10^6) = 6 \times 10^6$ computations.

Other properties of the Poison clock activation regime

- Over T periods, the mean number of activations per agent is T.
- The variance of the number of activations per agent is also T.
- Skewness and kurtosis are both 1.

The number of agents, n, that are active in each period is a random variable having a probability mass function:

$$f(n; N) = \frac{e^{-N} N^n}{n!}. \tag{1}$$

For large N this function assumes a Gaussian shape.

Preferential Activation The final activation regime we will briefly mention is Preferential Activation (see Page 1997). Agents can use resources to 'buy' activation time. this method has as a feature that activation is relatively costly, so only the agents who are successful can afford to buy more time and interact more often. This resembles the phenomenon of cumulative advantage, which is similar to the idea of preferential attachment in networks, and therefore this activation regime may lead to a scale-free distribution of the agent activation times.

Multi-role activation If agents have multiple roles for different forms of behavior (i.e., a household agent has a different role for consuming, working, trading, investing) then the problem becomes how to activate these multiple roles for each single agent. The activation regime now becomes an intra-agent problem. When an agent is activated by the large-scale activation regime its internal clock tells it that it has to activate its internal activation regime to determine what action to take next.
The agent may use a Poisson Clock Activation regime to chose between consuming and trading, or between working and searching for a new job. Alternatively, all the agent's internal rules get activated at once when the agent itself gets activated by the external activation regime, and all the different roles of the agent are acting out at simultaneously. This implies

that an individual agent is active on different markets simultaneously. Another possibility is to first activate all agents on a particular role and then repeat for all roles. This means that first every agent goes to work, then all agents consume, then all invest, etc. etc.
Since it is behaviorally not very credible to have such a *serial execution* of agent roles, it may be more pragmatic to use an *asynchronous parallel execution* instead, by giving each agent its own *'processing thread'* by which it decides which role it executes. This idea connects nicely to the notion of agents having autonomy and the ensemble of agent activations now becomes a *multi-threaded process*, i.e a parallel computation.

4.1.3 Agent activation in the X-agent framework

In EURACE we will make use of the X-agent framework FLAME (see Holcombe et al., 2006). It is clear that the FLAME framework uses parallel computing. Therefore, in principle, all agent activities within the EURACE simulator will be based on a fully asynchronous, parallel activation regime. But for some economic modelling purposes it may be necessary to use serial (non-parallelisable) operations. Almost all activities in the EURACE model are event-based, and depend on messages being send from one agent to another. The activities of an individual agent – its functions – depend solely on the messages it has received and the messages it is holding in its internal memory. Hence all the function dependencies between the agents are internalised through the use of messages, instead of through the functions themselves. If an activity of agent A requires information that is encapsulated in a message send by agent B, then agent A has to wait until agent B sends the message. Agent A actually has to retrieve and read this message before he or she can start the activity. However, some activities are not event-based but clock-based. Then a central clock is being used, and in the X-agent framework this will entail a centralized message being broadcast to all agents, telling them that a certain period has passed.
Concerning the issue of multi-role activation discussed above, the EURACE agents will be active on different markets and at different time-scales, thereby separating their multi-role activities in time without the need to have an internal time schedule to activate the different roles. In general, these roles will be placed in different market contexts and the activation of agents in a particular market context are then determined by the function dependencies. For example, in the context of the labor market, the only roles that are active are the *'employee role'* of the household and the *'employer role'* of the firm. The other roles for the same agents (consumer and producer), are not active in this market context, so there is no need to define any function dependencies between the functions of agent roles in different contexts. That means that different roles for the same agent can in principle be active simultaneously on different markets.

4.2 Time budgets

All processes in an agent-based model take time: decision-making, searching, information gathering, information dissemination, production and trading. This implies that agents have a limited amount of time to take decisions and perform tasks. In other words, the agents have a time-budget and are time-constrained.
The notion of agents having their own processing thread fits nicely here. It connects to the idea of Preferential Activation, where agents can buy *processing time* for their own thread to be activated. Agents who are more active than others may have a substantial advantage over their competitors, and may even be more likely to be active in the future due to the additional gains they can obtain when active.

4.3 The Central Clock

The issue of a central clock is important in relation to the global agent activation schedule and an agent's time budget. The central clock will be used to notify agents of the passing of time in the model, so that they can take this into account in their decision-making processes. How the central clock will be implemented is a computer engineering question, but how it will be used is an economic modelling issue.

4.4 The choice of a basic unit of time

Any computational description of economic activity requires a division of continuous time into discrete unit periods. For the EURACE simulator, we use the business day as the basic unit of time. This means that all processes, activities and decisions are expressed in discrete multiples of business days.

5 Interaction structures

[Note: This section on Interaction Structures has considerable overlap with Section 6.3 on decentralized exchange mechanisms. Topics concerning the matching protocols and the pair-wise matching algorithms fit better here, since they describe how the agents are paired and how trade links are formed.]

This section highlights different topologies for the interaction and trade relations between agents. In general, we can talk about the following types of interactions:

- Agent-agent interactions.
- Agent-environment interactions.

The next subsections provide details for each type of interaction structure.

5.1 Agent-agent interactions

Direct agent-agent interactions play an important role in everyday economic activities. Since a geographical structure has a most direct analogue in spatial networks, such local agent-agent interactions are easily implemented in ABMs by placing the agents on a grid. Information concerning the physical location can then be made part of the agent's internal memory, as is all local information in the X-agent framework.
Social network interactions are similarly represented by having the agents hold a list of pointers to other agents to represent their local neighborhood. Agent-agent interaction in the geographical network or the social network can be local or global, and direct or indirect. Any particular agent can be active in multiple network structures at the same time, e.g. an agent can be moving on the spatial grid while at the same time maintaining a friendship network. Theoretical studies of social networks have mostly focussed on specific network structures such as lattices or random graphs, for reasons of analytical tractability. In ABMs a more flexible approach can be taken, for instance by considering graphs with a non-constant, non-uniform degree distribution.
Another interesting possibility is to consider different layers of networks, in order to model the various relationships between agents in different market contexts:

- A network of buyer-seller relationships on the goods markets (loyalty, reputation).

- A network of employee-employer relationships on the labor market.
- A network of social relationships.

5.1.1 Interaction networks

To specify the interaction network we could use:

- Regular networks, i.e. lattices;
- Random networks;
- Small-world networks.

To deal with the network formation we have to consider the degree distribution of the links per node in the network:

- For regular lattices all nodes have the same degree k: all agents have the same number of links. For example, in the Schelling model all agents have a maximum of 8 neighbors (this may not be credible behaviorally).
- For networks with a constant but non-uniform degree distribution the number of links can be agent-specific.
- For networks with non-constant degree distribution the agents chose which new links to form, which old links to maintain, and which old links to sever (this may be the most credible behaviorally).

5.1.2 Network characteristics

Terminology. Network theory is based on graph theory, which has a somewhat different terminology. In graph theory the basic element is called a *vertex* (vertices) and the connections are called *edges*. In network theory the basic elements are called *nodes* and the connections between the nodes are called *links*. We will consistently use the network theory terminology for describing networks.

Nodes. Nodes are the basic network elements: $i \in N$, $N = \{1, ..., n\}$.

Network. A network is a collection of unordered pairs of nodes $G = \{..., (i, j), ...\}$, where $i, j \in N$.

Links. Two nodes i and j are linked if $(i, j) \in G$.

Directed links. $i \longrightarrow j$.

Undirected links. $i \longleftrightarrow j$.

Weighted links. $i \overset{0.2}{\longleftrightarrow} j$.

Connected set. A collection of nodes and links is said to form a ***connected set*** if for every pair of nodes i, j there exists a sequence of links $\{(i, i_1), (i_1, i_2), ..., (i_\nu, j)\}$ connecting node i to node j.

Network components. A network component is defined as a subset of nodes $N_i \subset N$ that forms a connected set.

Connected and unconnected components. A network consisting of a single network component is called ***connected***. A network with unconnected components is called ***disconnected***.

Component size. The component size is the number of nodes in a network component. The total number of links within a component does not scale linearly with size. The maximum number of links to form a completely connected component is $n(n-1)/2$.

Path. A ***path*** from node i to node j through the network is defined as any sequence of links $\{i, i_1, ..., i_n, j\}$ from i to j.

Path length. The ***path length for node*** i, ℓ_i, is defined as the minimum number of steps from node i to reach any other node in the network.

Diameter. The ***diameter*** d of a network is defined as: the longest path length, maximized over all nodes, $d = \max_{i \in N} \ell_i$.

Characteristic path length. The ***characteristic path length*** ℓ of a network is defined as the sum of all per node path lengths, averaged over all the nodes, $\ell = \frac{1}{n} \sum_{i \in N} \ell_i$.

Path lengths are only well-defined for networks that are connected. For networks that have unconnected components the shortest path length is infinite, hence the characteristic path length is also infinite. The complete network in Figure 1a has a characteristic path length equal to 1. The disconnected local network in Figure 1b has an undefined or infinitely long path length.

Shortcuts. Shortcuts are network links that reduce the shortest path from a particular node to another node.

Bridges. A bridge is a special type of network link that connects two network components. If the bridge is severed the two components become unconnected and the network becomes a disconnected network. Note that a shortcut is *not* the same as a bridge, or vice versa, although the creation of a bridge does reduce the shortest path length from infinite to finite length.

Degree. The ***degree*** k_i of node i is the total number of links of node i. In a directed network each node has both an in-degree and an out-degree, corresponding to the number of links going into the node and out of the node, respectively.

Average degree. The average number of links per node: $< k >= \frac{1}{n} \sum_{i \in N} k_i$.

Degree distribution. The distribution of degrees k_i over all the nodes $i \in N$.

Clustering coefficient. If node i has k_i immediate neighbors these define a sub-network that can have at most $\binom{k_i}{2} = k_i(k_i - 1)/2$ links. Define C_i as the ratio between the actual number of links in the network component and the maximum number of links that can possibly exist:

$$C_i = \frac{\text{actual number of links}}{\text{number of possible links}} = \frac{k_i}{k_i(k_i - 1)/2} = \frac{2}{k_i - 1}.$$

The clustering coefficient C is the average of C_i over all the nodes in the network

$$C = \frac{1}{n} \sum_{i \in N} C_i.$$

5.1.3 Network structures

Figure 1 shows four distinct network structures: a Complete network, a Local disconnected network, a Local connected network and a Small-world network (figure reproduced from Wilhite, 2001).

Complete networks. In a complete network every agent can trade with every other agent. All nodes are completely connected, see Figure 1a.

Local disconnected networks. In a local disconnected network there exist disconnected network components, see Figure 1b.

Local connected networks. In a local connected network the network components have minimal overlap. For example, two agents per group, see Figure 1c.

Small-world networks. A small-world network can be constructed by taking a local connected network and then adding a few additional links that connect a node to another distant node, that is not part of the local group. These links are ***shortcuts*** through the network, see Figure 1d. A small-world network has a high clustering coefficient C and a low characteristic path length ℓ.

Random networks. For a random network, the clustering coefficient C is equal to the probability of any two nodes being connected, p, which is the same for any node and is equal to $< k > /N$, where N is the number of nodes in the network and $< k >$ is the average degree, i.e. the average number of links per node. The characteristic path length for a random network is $\ell = ln(N)/ln(< k >)$. ℓ therefore scales with the logarithm of N, so it increases at a much slower speed than the size of the network.

Scale-free networks. Barabasi and Albert (1999) rediscovered a result by de Solla Price (1965, 1976) on the growth of networks. De Solla Price (1965) studied scientific citation networks and discovered the effect of ***'cumulative advantage'***: papers which are cited more are more likely to be cited more often in the future. Barabasi and Albert (1999) studied the growth of the World Wide Web and the network of hyperlinks and introduced the term ***'preferential attachment'*** for the same phenomenon: websites which are hyperlinked more, are more likely to be hyperlinked more often in the future. Both are based on work by Simon (1955) who showed that power laws arise in the income distribution when there is a ***'rich-get-richer'*** phenomenon: people who are richer are more likely to be even richer in the future (see also Newman, 2003).

Scale-free networks are based on the notion of having nodes in the network that function as hubs. The network contains only a small number of nodes with a large number of links, and a large number of nodes with only a few links. The way to generate such a network using preferential attachment is nothing more than stating that growth is proportional to cluster size. With preferential attachment, new links are formed as follows. Each node in the network has a probability of receiving a new link that is proportional to the number of links it already has. Therefore large clusters grow faster than small clusters. The probability distribution of the node connectivity $P(k)$, i.e. the probability that a node in the network is connected with k other nodes, has an exponential drop-off and a characteristic size of $< k >$, equal to the network's average degree. This means that $P(k)$ is free of scale and follows a power-law. The resulting plot of $C/ < k >$ against N, on a log-log scale, should appear as a straight line. The quantity $C/ < k >$ is the clustering coefficient divided by the average degree for the network as a whole. And this is precisely what scale-free networks show: a scale-free distribution of the average number of links per node, i.e. the average degree distribution over the entire network is a power law.

Network type:	Complete	Local disconnected	Local connected	Small-world
Nr. of groups	1	50	50	50
Nr. agents per group	500	10	11	11
Nr. of overlaps per group	no	no	yes	yes
Nr. of cross-overs	0	0	0	5

Table 1: Network topologies.

5.1.4 Network construction

Wilhite (2001, p. 55) proposes to construct these networks using four parameters, which can be set by the user:

1. The number of trade groups;
2. The number of agents within each trade group;
3. Whether or not trade groups overlap;
4. The number of shortcuts in the network.

Table 1 lists the parameter settings for the four distinct types of network topologies.

Construction of small-world networks by random rewiring of links. A way to build a small-world network is to start with a regular lattice of N nodes, each connected to k neighbors, and randomly rewire each link with probability p, excluding self-connections and duplicate links. For $0 \leq p \leq 1$ a sequence of networks is obtained of which the regular lattice ($p = 0$) and the random network ($p = 1$) are extreme cases.
A regular lattice has a high clustering coefficient C and a high average path length ℓ. A random network has a short average path length ℓ and a low clustering coefficient C. A small-world network has at the same time a high clustering coefficient C and a short average path length ℓ.

5.1.5 Trade on networks with bilateral trade

Below we discuss and interpret the results of Wilhite (2001). One central question concerns the speed of convergence towards equilibrium. The complete network is very efficient in converging to an equilibrium price distribution in the sense that a small number of trading rounds is needed and the total number of trades required are relatively limited. But this comes at a cost. The search process for an agent to find a suitable trading partner in the complete network is very costly. With every agent negotiating a unique 'local market price' that is specific to each agent-pair, every trade between a pair of agents is the result of an extensive search among the entire agent population. So even though the actual trading rounds are short, the search-and-negotiation round can take a very long time.
With local disconnected markets, trade is also settling down rather quickly to an equilibrium distribution of prices, but since markets are locally isolated there is a large deviation in the price distribution. The search costs during the negotiating phase are considerably smaller than for the complete network. Total searches to find a trading partner in the local disconnected network are far less since there are fewer agents in each network component. Summarizing, the

disconnected markets are inefficient because potentially mutually beneficial trades are left un-traded. This leads to large price dispersion.
For the local connected markets the searching-for-trade-partner phase is again costly. The total number of searches required before a mutually beneficial trade can be found is considerably large. Also, the total number of trades required to reach an equilibrium distribution of prices is very high. Summarizing, this network structure is also not very efficient in finding an equilibrium price distribution. The reason for this is in the local connectedness of the network structure. Since the local neighborhoods are small, the number of searches required to find potential trading opportunities ***locally*** are small, but in order to exploit all possible opportunities for trade ***globally*** requires a large number of trading rounds. This refers to a basic trade-off in local connected networks: information diffusion is locally very fast, but globally it is very slow.
The final step is to consider the local connected network and add some shortcuts between the local agent groups to obtain a small-world network. Now the results are different. All trade is local and the information is locally distributed across the network, but the global diffusion rate is very high due to the shortcuts in the network structure. The number of searches required in every trading round to find suitable trading partners was reduced by 40% in comparison to the complete network. Only 1% of the agents needed to be a cross-over agent providing access to some distant non-local market to produce this result. This refers to one of the basic features of small-world networks: both local and global rates of information diffusion are high.

Considering these results, we obtain the following ranking in terms of network information efficiency (measured as a trade-off between price dispersion, total number of trades and total number of searches): Complete network > Small-world network > Local connected network > Local disconnected network.

5.1.6 Small-world networks

Two main characteristics of small-world networks are important: the characteristic path length ℓ and the component size or group size n_v.
In the bilateral trade network of Wilhite (2001), the group size is connected to search costs across the local component of the network, while the path length is connected to the total number of trades that is required before all potentially beneficial trades have been exhausted. If we regard the links as communication channels between agents then search costs for a single agent increase with group size: larger groups incur greater communication costs to the individual. Hence, if a population is divided into smaller subgroups then the communication costs decrease. But for the group as a whole the total number of searches scales as $n_v^2 - n_v$, where n_v is the group size.That is, the total number of communication messages required in order for every agent to find all potential trading partners and to extract all trades that are mutually beneficial, is equal to $n_v^2 - n_v$.
In the small-world network of Figure 1d, the average path length becomes shorter as more shortcuts are added. This shortening of the average path length occurs at a faster pace than the increase in the group size (as a shortcut is formed, the two agents on both sides of the shortcut each become part of two groups, hence the average group size increases). Therefore, the total number of searches in the group increases slower than the total number of trades and this is beneficial for the network's efficiency with respect to information dissemination.
The reason why shortcuts are so important in network formation is that it has the potential to shorten the path length not only for the two agents directly involved in building the link, but

also for almost every other agent in the entire network. If adding a link indeed reduces the path length per node then it also reduces the shortest path length for the network as a whole, since this is the average of all the per node path lengths.[1] As a consequence, the characteristic path length of the network is also reduced. This has the added benefit that the total interactions needed to communicate with all other nodes becomes less costly, i.e. it requires fewer communication messages.

Summarizing, a small-world network has the property that it has a short characteristic path length combined with a high clustering of nodes, which cluster together to form locally well-connected network components. This result has led to the belief that small-world networks make for robust and efficient network design.

5.1.7 Empirical results on real-world networks

Empirical studies have provided evidence that real-world social- and economic networks share some basic characteristics with small-world networks. We cannot say with absolute certainty that real-world social networks are small-world networks, but they can be modelled as such. Examples are the following:

- links between movie actors (the famous Kevin Bacon Network).
- links between Internet servers.
- links in high-school friendship networks.
- marriage links between rich families in 15^{th} century Italy.

There are many good surveys on empirical networks, and we will not try to give an exhaustive overview here. The interested reader is referred to the survey by Newman (2003) and the references therein.

5.2 Agent-environment interactions

Agent-environment interactions deal with the methods used by agents to sense their environment. How well agents can respond to their environment depends on what signals are transmitted to the agent, and what signals the agent is broadcasting to the environment. Also, another important aspect might be the agent's confidence in the fidelity of those signals. Agents should have sufficient ***cognitive abilities*** to perceive and interpret patterns occurring at the macro-level. This refers to Sargent's recommendation "for building little econometricians into models" (see LeBaron et al., 1999, p. 1488, and Sargent, 1993). In order for this to be the case the model requires the following features (see also Dessalles and Phan, 2004, p. 9):

1. Agents have the ability to use 'probing tools' to probe the macro-state (e.g. every node in the cluster publishes a newspaper). That is, agents have access to a set of detectors to detect the parameters at the macro-level.
2. Agents can describe the epiphenomena they observe in a language other than the one which is being used to describe the process in which they take part. The agents should be able to form idiosyncratic models of their environment.

[1] Adding a shortcut may not reduce the path length for some particular nodes, but it certainly does not increase it either.

3. Agents have the ability to change their behavior, i.e. to adapt, in response to the detected epiphenomena. Also their subjective models should be adaptable.

What we then have is a model with sentient, adaptive agents who are aware of their environment and who can make use of detectors that enable them to form concepts about their environment. In addition, they are able to formulate models about the behavioral repertoires of the other agents. This yields a multi-agent system with social agents.

6 Market types

The purpose of this section is to provide a *catalogue* of different market protocols that have been studied in the theoretical literature. It is a listing of the choice set for the EURACE unified modelling framework.

Markets are interfaces between market participants. The main role of markets is to facilitate the coordination of economic activities. As a definition of what is a market, we can take a very general definition: "A market is any context in which buyers and sellers exchange a commodity."

Although some economic transactions may occur between economic actors directly, most transactions must pass through a market interface. In the sections that follow, we will describe canonical models for a number of market mechanisms:

- Centralized exchange mechanisms (single intermediator).
- Semi-centralized exchange mechanisms (multiple intermediators).
- Decentralized exchange mechanisms (no intermediator).

6.1 Centralized exchange mechanisms

A centralized exchange mechanism implies a single, centralized market for all traders who want to participate in the exchange of a certain commodity or asset. It does not necessarily mean that there exists one single centralized market for all commodities in the economy to be traded simultaneously. For every commodity in the economy there can exist a separate – but fully centralized – market on which all trades in the commodity occur.
We start our market catalogue with a description of some auction mechanisms because they are simple and well-defined economic environments. Auctions are used for many economic transactions, for example by governments to sell treasury bills and procurement contracts, by firms to sell subcontracts, and by private individuals to buy and sell antiques, used products, artwork, etc., etc. Other examples include the auctioning of spectrum licenses in the US governments FCC Spectrum Auction (see Klemperer, 2002a; Binmore and Klemperer, 2002; Klemperer, 2002b), and the electricity market in Sweden (Swedish Competition Authority, 1996). For much more information on auction theory and its applications, see the survey by Paul Klemperer (Klemperer, 2000) on which this section is based.

6.1.1 Single-sided auctions

A single-sided auction is an auction mechanism in which only one side of the market is active. Below we distinguish between: ascending vs. descending, sealed-bid vs. open-bid, first-price vs.

second-price, and single-unit vs. multi-unit auctions. All of these cases will be single-sided auctions in which the bidders are either buyers or sellers for the object(s) that is being auctioned.

Ascending vs. descending auctions. The first characteristic of an auction mechanisms is whether it uses ascending or descending bids. An auction with ascending bids is also called an open, oral or English auction. An auction with descending bids is known as the Dutch auction (user in the sale of flowers in the Netherlands). A model that is often used is the Japanese auction, where the price rises continuously while the bidders gradually quit the auction. Bidders who have quit are not allowed to re-enter the auction.

Sealed bid vs. open-bid auctions. In a sealed-bid auction no information is given to the market participants about the outstanding bids. Bidders independently submit a single bid. In an open-bid auction all market participants know what is the current best bid and/or they know all the bids that have been made so far.

First-price vs. Second-price auctions. In a first-price sealed-bid auction the highest bid wins and the highest bidder has to pay his bid, or "first price". In a second-price sealed-bid auction the highest bid wins as well, but the object is sold to the highest bidder at the *second*-highest bidder's bid, or "second price".
In the theoretical literature on auctions the second-price sealed-bid auction (also known as the Vickrey auction by economists) is well-known for its efficiency properties, which are mainly due to the revelation principle: bidders make truth-revealing bids that reveal their true value for the item, but not more than that (see Klemperer, 2000). Truth-telling is a dominant strategy equilibrium.

Discriminatory price vs. uniform price auctions. In auctions with discriminatory pricing each bidder pays her own bid price. With uniform pricing, all bidders pay the same price, which can be the first-price or the second-price.

Bid-improvement rules A bid-improvement rule is a rule that states that any new bid must improve on the current best bid. For a sell bid this means that it must be at a price at least one increment below the currently best selling price, and for a buy bid it means it must be at a price at least one increment above the currently best buying price.

Some important remarks. The ascending auction with open bidding (and private values) is also called an "open second-price auction". Since prices are ascending, every bidder stays in the auction up until her private value for the object is reached and then drops out if the price is increased further. The bidder with the highest private value will therefore stay in the auction until the very end, while the second-highest bidder drops out at the last moment. With only one bidder left the auction stops and the bidder buys the object for the current price. This is equivalent to having the highest bidder paying the second-highest price in a second-price sealed-bid auction.
The descending auction with open bidding (and private values) is also known as an open first-price auction. Since prices are descending gradually, it is the first bidder who calls out that she accepts the price who gets to buy the object. This is equivalent to the outcome of a first-price sealed-bid auction.

The equivalences only hold in the private value-model, but not for the model with common-values, i.e. where a bidder's value for the object depends on what they learn during the auction about other bidder's values by observing them quitting the auction.

Finally, there is no formal distinction between a *normal auction* in which the auctioneer is the seller and the bidders are the buyers, and a *procurement auction* in which the auctioneer is the buyer and the bidders are the sellers, who have costs for supplying the objects that are sold.

Example 1. eBay auctions. The eBay system (1992) is a electronic auction site for collecting bids and offers through the internet. The mechanism by which market participants receive public information has gone through several changes over the years.
Consider an eBay auction for a single unit of a single item. The auction is a second-price, sealed-bid auction, meaning that the highest bid wins but the person who entered the highest bid only has to pay the second-highest bid. The public information consists of a list of the history of all bids, *excluding* the current best bid for an item. The person who has entered the current best bid does not see his/her bid on the history list, nor do any of the other participants.
Bajari and Hortaçsu (2003) provide the following description of the eBay auction mechanism:

> 'The bidding format used on eBay is called 'proxy biding'. Here's how it works. When a bidder submits a proxy bid, she is asked by the eBay computer to enter the maximum amount she is willing to pay for the item. Suppose that bidder A is the first bidder to submit a proxy bid on an item with a minimum bid of $10 (as set by the seller) and a bid increment of $.50. Let the amount of bidder A's proxy bid be $25. eBay automatically sets the highest bid to $10, just enough to make bidder A the high bidder. Next suppose that bidder B enters the auction with a proxy bid of $13. eBay then raises bidder A's bid to $13.50. If another bidder submits a proxy bid above $25.50 ($25 plus one bid increment), bidder A is no longer the high bidder, and the eBay computer will notify her of this via e-mail. If bidder A wishes, she can submit a new proxy bid. This process continues until the auction ends. The high bidder ends up paying the second-highest proxy bid plus one bid increment. Once the auction has concluded, the winner is notified by e-mail. At this point, eBay's intermediary role ends and it is up to the winner of the auction to contact the seller to arrange shipment and payment details.' (Bajari and Hortaçsu, 2003, p. 329-30)

Multi-unit, single-item auctions. In addition to the single-unit, single item auctions described above, there also exist multi-unit, single-item auctions in which multiple units of the same type of item (i.e. object) are on auction.
The first- and second-price auctions for single-unit auctions can be generalized to the k-unit case: 'Suppose k units are offered for sale. In a generalized second-price auction a uniform price is set at the level of the highest rejected bid [i.e., the second price bid]. The highest k bidders receive one unit each and pay the uniform price.' (Wolfstetter, 1999, p. 207) 'In a generalized first-price auction, the highest k bidders are awarded the k [units] and each pays his own bid. Therefore, the generalized first-price auction involves price discrimination.' (Ibid.)
When the bidders are allowed to only procure a single unit of the k units on auction, and each unit is awarded to the highest bidder, then all auction rules are revenue-equivalent. This does not generalize however when bidders can buy multiple units.

Procurement auctions. Procurement processes for the procurement of government contracts are usually in the form of a first-price, sealed-bid auction. Multi-unit auctions are mostly used in financial markets, e.g. by the US Treasury to sell marketable bills, notes and bonds. These items are sold in more than 150 regular auctions per year, using a sealed-bid, multiple-price auction (see Wolfstetter, 1999, p. 208).

Multi-unit, multi-item auctions: Package bidding. In case of multiple units of multiple items there is the problem of complementarity: some combinations of the units of different items are useful while others may not be useful. For this reason it is sensible to allow bidders to bid for bundles of commodities instead of for the items separately. This procedure is called package bidding. Multi-unit auctions are also called ***combinatorial auctions*** as it refers to the auctioneer having to solve a combinatorial optimization problem to find an optimal allocation given the bidders' package bids.

Simultaneous Ascending Auction. The most important new auction design is the Simultaneous Ascending Auction (SAA). This is a fairly natural extension of the basic ascending auction to the multiple objects case; the bidding remains open on all the objects until no-one wants to make any more bids on any object (see Klemperer, 2000, p. 89).

6.2 Semi-decentralized exchange mechanisms

The basic characteristic of a semi-decentralized exchange mechanism is that there are multiple intermediators competing to mediate the trades between buyers and sellers. A stock exchange with a limit-order book is such a semi-decentralized mechanism (since there are multiple order-books, one per asset traded). The market participants are free to chose the intermediary with whom they wish to have a client-server relationship. It is not a fully decentralized system, because there is no local decentralized trading among the clients themselves, since they have to pass the information about their orders *through* the market interface.

6.2.1 Double auctions

Double-sided auctions are auctions in which both sides of the market are active. These can be either one-shot auctions that only happen at set times, as used for example in the electricity markets, or they can be continuous double-auctions in which orders arrive constantly and continuously.

6.2.2 Batch auctions - Clearinghouse mechanism

The clearinghouse mechanism is a form of market organisation that is used as the default for electricity markets (see Swedish Competition Authority, 1996, Gonzalez and Basagoiti, 1999, Guerci et al., 2005). After all the market participants have submitted their orders, all bids and offers are aggregated into market demand and supply schedules as shown in Figure 2. Then a marginal price is found at which the market clears. Only orders that have been entered at prices better than the marginal price are executed. This means that all buying orders submitted at or above the marginal price are executed, as well as all selling orders at or below the marginal price. In Figure 2, only the orders on the left-side of the market, i.e. to the left of the market clearing point, are executed.[2] The clearinghouse mechanism is also called a batch

[2] This implies that the marginal price cannot be interpreted as a market clearing price or an equilibrium price, since there are orders left unsatisfied to the right of the market clearing point. Some may argue that it *is* an

auction, due to the fact that orders are first collected and batched before any transaction occurs.

6.2.3 One-shot double auctions - Electricity Markets.

The electricity markets are a case in point when it comes to market design. In 1995 the Swedish government decided as one of the first to liberalize its market for electricity starting on January 1st, 1996. The Spanish government followed and on January 1st, 1998 the Spanish Power Exchange Market became operational. The description below is of the Spanish market and is based on Gonzalez and Basagoiti (1999). All market rules for the electricity market auctions are public and available to any agent or potential agent.[3]

Market participants. There are five types of market participants in the Spanish Power Exchange Market:

1. Energy Generator companies: all generators with a production capacity higher than 50MWh are obliged to sell energy on the Energy Market.
2. Energy Distributor companies: sell energy at the regulated tariff, are not allowed to sell to qualified consumers.
3. Energy Reselling companies: allowed to sell to qualified consumers, or re-sell to other re-sellers.
4. Qualified consumers: minimum annual consumption of at least 1GWh.
5. External agents: generators, retailers, or consumers from other countries.

Consumers can buy energy in four distinct ways:

1. From an Energy Generator company by means of a physical bilateral contract.
2. From an Energy Distributor company at the regulated tariff.
3. From an Energy Reselling company at a reselling price.
4. From the Energy Market directly.

Market organisation. The market organisation is illustrated in Figure 3. The exchange structure consists of a Day-Ahead Market during a single session held daily, and an Hour Ahead Market during five separate sessions held afterwards.

---

equilibrium price precisely for the reason that all the unsatisfied orders are placed at *higher* than market-clearing prices, and that therefore *at* the marginal price there are no more unsatisfied orders to fulfill. However, this statement is antithetical to the notion that an equilibrium price is defined as: "that price at which there are no more traders willing to trade, even at higher prices."

[3] It is downloadable from a public website: `http://www.mercaelectrico.comel.es`.

Day-Ahead Market. The Day-Ahead Market (DAM) is operated by the Market Operator (MO). The DAM is basically a double-auction for the exchange of contracts promising the delivery of electricity one day ahead, for each of the 24 hour-slots of the subsequent day.[4] During each DAM the activity consists of bidding simultaneously on the 24 one-shot double-auctions that are in operation. Hence, the market for electricity is a combinatorial auction. Bids can be entered between 8.30am-10am.

Bidding. Each hourly slot is subdivided into 25 smaller parts, i.e. into 2.4 minute intervals. A day has $24 \times 25 = 600$ energy blocks in total. On the daily market sellers and buyers can present only one bid for each energy block. At this stage, the sellers are the generator companies and the buyers are all the other market participants.
There is a bid-improvement rule: 'For selling bids the price of the blocks need to be increasing with the energy bided on the hour, and for purchasing bids the price of the blocks need to be decreasing with the energy requested on the blocks.' (ibid. p. 3)

Optimization. Between 10-11am the MO collects all bids and starts the process of combinatorial optimization. When this procedure has finished this results in 24 marginal prices, a schedule for the production and delivery of electricity by each generator for each hour-slot, and commitments to purchase this electricity by the buying parties at the marginal prices as set by the MO. This constitutes a so called ***'unconstrained solution'***, since the network capacity constraints have still to be taken into account. It is the task of the Systems Operator (SO) to solve generation/load imbalances on the grid. The 'unconstrained solution' is transmitted by the MO to the SO (before 11am) who then checks for the technical viability of the daily energy schedule. The SO, in co-operation with the MO, then solves for a ***'constrained solution'*** which now yields a ***'technically viable daily schedule'*** (before 2pm).

Hour Ahead Market - Intra-Day Market. After the Day-Ahead Market closes, some generators or distributers may want to make some adjustments to their delivery/production commitments in order to satisfy their bilateral contracts. This is done on the Hour Ahead Market. Of course, all previous transactions and commitments made in the Day-Ahead Market are firm and cannot be undone. That is why the Hour Ahead Market is merely an adjustment market. The HAM is held in five separate sessions for time horizons of 28 hours, 24 hours, 19 hours, 14 hours and 10 hours, respectively. Participation in the HAM is completely voluntary and unrestricted, provided that buyers have participated in the corresponding daily market session:

> 'The cited purchasing bidders in the daily market, and those who have purchased power through a physical bilateral contract, shall only be allowed to participate with respect to the hourly scheduling periods included in the intra-day market session if they participated in the corresponding daily market session, or, if they have bought power through physical bilateral contract, they did so on the day targeted for the intra-day market session.' (Spanish Market Authority, 2001, p. 53)

On the Hour Ahead Market, the SO request 'secondary regulation needs', asking the Generator Companies for ancillary services to resolve any technical problems. The process of

[4] A provision is made that there will be twenty-three or twenty-five periods on days when clocks are changed to go on or off Daylight Savings Time.

transmittal of bids for these ancillary services takes place at the existing marginal prices of the daily market, and subsequently the SO assigns these ancillary services to the Generator Companies. This process ends before 3.30pm.[5] Once new marginal prices have been determined for the $\{28, 24, 19, 14, 10\}$-Hour-Ahead-Markets the SO checks again for viability. All transactions that violate a technical viability constraint are cancelled by the SO afterwards, and a final 'technically viable daily schedule' is published by the SO. Before 4pm the final solution is communicated to all sellers and buyers, giving them only the schedule for their private commitments.

6.2.4 Continuous double-auctions - Limit-order markets.

Many of the world's financial markets are organised as continuous double-auctions (CDAs). It is an efficient and transparent way to collect the buy and sell orders for a financial asset such as stocks, options, futures, etc. Below we give a canonical description of a limit-order market, which is a particular instantiation of the CDA mechanism.
A limit-order market consists of orders entered as tuples of prices and quantities (p, q), where p is the 'limit price' and q is the 'limit quantity', or the size of the order. The limit price and limit quantity are the 'worst' price and 'worst' quantity, respectively. The price p is the *maximum* unit price at which a buyer is willing to buy the asset, or the *minimum* unit price at which a seller is willing to sell the asset. For both cases, q is the maximum quantity a trader is willing to trade.
Sellers announce sell-orders by stating a *limit-ask* price (p^a). Any transaction price $p \geq p^a$ is also accepted. The seller wants to sell the maximum quantity q^a. A limit-order to sell by agent i is given as: (p_i^a, q_i^a). Buyers announce buy-orders by stating a *limit-bid* price (p^b). Any transaction price $p \leq p^b$ is also accepted. The buyer wants to buy the maximum quantity q^b. A limit-order to buy by agent i is given as: (p_i^b, q_i^b).
We have still to explain what it means for a buy and a sell order to 'match'. A tuple of buy/sell orders $((p^b, q^b), (p^a, q^a))$ matches when $p^a \leq p^b$, i.e., the bid-price is higher or equal to the ask-price. A market transaction occurs at the price at which the first order arrived at the market, i.e. it is the price of the order with the longest waiting-time. The transacted quantity is the minimum of the two limit-quantities: $q^{trade} = \min\{q^a, q^b\}$.

Generically, a limit-order market has the following properties (for an illustration, see Figure 4):

1. Limit-orders are allowed to arrive at the market continuously and independently.

2. Limit-orders are entered into the Limit-Order Book (LOB) instantaneously the moment they arrive and the orders receive a time-tag.

3. When a buy and sell order are 'matched' a transaction occurs and the executed orders are (partially) removed from the LOB. The remaining quantity of a partially executed order remains in the LOB at the same limit-price.

4. The order-matching and order-execution algorithm is implemented by the market authority. All information necessary for traders to participate in the market is available (freely or at some costs).

[5]The geographical structure of the energy network in Spain is such that it is possible to use this simple procedure to solve for the technical restrictions. For countries which do not have such a convenient round shape it could be much more difficult to solve the technical network constraint problems. For example, Italy, which has an elongated shape, uses two markets: one for the North and one for the South.

5. The order flow in the limit-order book is not accessible to all market participants. Partial information on the order flow can be purchased.

Different order-matching and order-execution algorithms exist, depending on the rules that have been set by the market authority. Therefore no generic model exists for all financial markets, since it is very much dependent on the rules in use. However, there do exist some common rules that indicate how orders are ranked and how they are traded according to certain priorities:

1. Price priority: an order at a higher price transacts first.

2. Volume priority: if orders have the same limit-price, the order with the highest volume transacts first.

3. Time priority: if orders have the same price and volume, the order which arrived first will transact first.

4. Tie-breaking rule: market dependent. Orders having the same price, the same volume and the same time-tag can be executed based on a market makers discretion.

Terminology. The standard terminology for limit-order markets is the following:

Current ask: the minimum of all ask prices currently in the order book, $A_t = \min_i\{p_i^a\}$.

Current bid: the maximum of all bid prices currently in the order book, $B_t = \max_i\{p_i^b\}$.

Bid-Ask spread: The difference between the current ask and the current bid, $S_t = A_t - B_t$.

Bid-quote: The price that gets quoted by a market maker as being the current bid.

Ask-quote: The price that gets quoted by a market maker as being the current ask.

Market price: the price of the most recent transaction. This can differ from the current bid and current ask price.

Market order: An order that is entered at the current market price p^M. The order need not transact immediately as it depends on the state of the LOB. In particular, a market order can have an expiration date, indicating how long it should remain in the LOB.

Some remarks. A limit-order book holds all the current orders for one particular asset being traded in the downstairs market. There is also the upstairs market where the market makers are trading amongst each others in order to close their positions. On multi-asset markets there are multiple limit-order books, one for every asset. The information that is available in all the LOBs is only available to the market makers and market specialists.

Conditioning orders. Just recently the literature on financial market microstructure has begun to study systems of multiple assets in which market makers are able to observe and condition on multiple order flows simultaneously (see Cespa, 2004). This has led to the consideration of cross-conditioning of orders: a trader may not only want to condition her limit-order on the price of the asset she is trading, but also on the prices of other assets. Another possibility is that traders may want to trade a vector of assets (i.e. a portfolio) simultaneously, and condition their orders accordingly. Such multi-price contingent orders consist of a specification of the parameters upon which the order execution should be conditioned.

In the case of single-price contingent limit-orders (the order only depends on the asset's own price) the market makers compete only for the order flow of the asset they are assigned to. In the case of multi-price contingent limit-orders, the market makers compete for each asset order flow. Consequently, the market makers can set prices conditionally on the vector of all order flows, and take into account all cross-order flow information about fundamentals. In the single-asset case, they only take into account the order flow of the single asset they are assigned to.

The general belief is that multi-price contingent orders will render the market more efficient: 'A mechanism which enables *simultaneous conditioning* of orders for different assets [...] would increase the information available to traders, improve value discovery and reduce volatility' (Amihud and Mendelson, 1991, p. 127, original emphasis). However, in Cespa (2004) this view is contested and it is shown that such cross-conditioning can deteriorate the efficiency of the price-discovery process.

6.3 Decentralized exchange mechanisms

Decentralized exchange mechanisms differ from fully centralized and semi-centralized mechanisms by the property that there are no intermediators. In this sense one can debate whether one can speak of 'markets' in the strict economic sense, that is, 'any context in which buyers and sellers come together to exchange a certain commodity.' In fully decentralized exchange mechanisms all information is locally distributed among the agents and there does not exist a market per economic good, but a market for every *pair* of agents.

When there are distributed local interactions of independent agent pairs who are each engaged in bilateral exchange we should redefine the meaning of a commodity to denote any economic good that is exchanged by a particular pair of agents. We can then define a commodity à la Debreu stating all physical characteristic $i = 1, ..., N$, state contingencies $s = 1, ..., S$, time of delivery $t = 1, ..., T$, and the current owner of the commodity $h = 1, ..., H$ and the buyer $k = 1, ..., K$. The quantity variable now has five indices: $x(i, s, t, (h, k))$. This deviates from the neoclassical formulation of markets in which commodities are owner-independent and all agents are anonymous, i.e. the ownership property does not matter. Also the location of the commodity has to be denoted in principle, but here the local market simply consists of the agent pair, so the location is suppressed. With bilateral exchange relationships, ownership becomes important and there is a price $p(i, s, t, (h, k))$ associated to every agent pair (or local market).

This specification of local markets leads to an explosion of the commodity space. In fact, each unit of any particular commodity could now be viewed as having its own local market and as a consequence having its own local price associated to the particular agent pair trading the unit. This means that markets become very thin and every bargaining processes is an auction between two traders for a single unit of a particular type of commodity that depends on place, time, and the identity of the owner and buyer.

The A4-model of neoclassical economics. The neoclassical research program rests on the A4-model of market interactions between Autonomous, Anonymous, Atomized Agents.

- Autonomy: agents have independent decision-making routines (not necessarily isolated from their economic environment or social context).
- Anonymity: all interactions are strict market interactions (no bilateral trade relationships).
- Atomization: agents have no market power; they are isolated entities that can be clearly separated from their environment.

As was noted above, bilateral exchange mechanisms require the relaxation of the anonymity property. Also the property of atomized agents (no market power) shall have to relaxed in the agent-based models that we consider in the EURACE project. The autonomy property will have to be maintained since it is an essential aspect of any agent-based model that the agents are autonomous decision-makers.

6.4 Bilateral bargaining protocols

In order to engage in trade agents have to perform a local search and use some bargaining protocol. This can be either bilateral or n-lateral, where $n > 2$. There exists a large literature on bilateral bargaining protocols, see Feldman (1973), Albin and Foley (1992), Bell (1997) and Wilhite (2001). Below we provide descriptions of these.

6.4.1 Edgeworth barter process

The bargaining rule in the Edgeworth bilateral barter process is that exchange takes place as long as trade is mutually beneficial for the agents involved in the exchange. Commodities are durable, hence the total stock at the beginning of the process is preserved. Basically, the process is a local gradient search for utility improvements, and a standard result from the literature is that one can find a Lyapunov function (see Axtell, 2005 for details). The process converges in finite time to an equilibrium point that is a Pareto Optimal allocation, although it may not necessarily be an equilibrium with respect to the initial holdings (due to the changes in the holdings of the agents along the trajectory).
A mutually beneficial exchange exists if two agents differ in their marginal rates of substitution between any two given commodities.[6] This statement can be reformulated in terms of the normalized gradients of the agents respective utility functions. Define:

$$MRS^i = \frac{\nabla U^i}{||\nabla U^i||}, \text{for all agents } i. \tag{2}$$

Then a mutually beneficial exchange exists as long as $MRS^i \neq MRS^j$, that is, the normalized utility gradients are not equalized and it is beneficial for agents i and j to trade (if they meet). At an equilibrium point all potential utility improvements have been extracted and the

[6] For agent i, the marginal rate of substitution between commodities a and b is defined as $MRS^i_{a,b} = \frac{\partial U^i}{\partial a} / \frac{\partial U^i}{\partial b}$.

gradients are equal. This is equivalent to the statement that the equilibrium price vector is equal to the normalized utility functions:

$$(p_1^\star, ..., p_n^\star) = \frac{\nabla U^i}{||\nabla U^i||} = \frac{\nabla U^j}{||\nabla U^j||}, \text{for all agents } i, j. \tag{3}$$

The Edgeworth exchange process proceeds as follows. Both agents have a marginal rate of substitution between two commodities a and b. They exchange commodities according to the exchange ratio $P = MRS^i_{a,b}/MRS^j_{a,b}$, which is equivalent to a 'price' when one of the commodities is taken as the unit of account. Taking commodity a as the unit of account, a has a relative price of $P_a(a) = 1$ and b has a relative price of $P_a(b) = \min\{MRS^i_{a,b}, MRS^j_{a,b}\}$. Agent i owns a and wants commodity b, while agent j owns b and wants commodity a. Agent i wants to use the exchange ratio $a : b = p^i_a : p^i_b$ while j wants to use the exchange ratio $a : b = p^j_a : p^j_b$.

The bilateral exchange ratio (or 'price') can be obtained using many price setting rules:

- Geometric mean: $p = (MRS^i(a, b) \cdot MRS^j(a, b))^{1/2}$.
- Arithmetic mean: $p = (MRS^i(a, b) + MRS^j(a, b))/2$.
- The minimum: $p = \min\{MRS^i(a, b), MRS^j(a, b)\}$.
- Any other rule:[7] $p = [\frac{(\partial U^1/\partial x_1)}{(\partial U^1/\partial x_2)} + \frac{(\partial U^2/\partial x_1)}{(\partial U^2/\partial x_2)}]/2$.

Example. Suppose we use the minimum rule: $p = \min\{\frac{p^1_a}{p^1_b}, \frac{p^2_a}{p^2_b}\}$. The goods are apples and bread. Agent i owns apples, but needs bread, while j owns bread but needs apples. Agent i thinks that 1 loaf of bread has the same value as 5 apples and has a marginal rate of substitution between apples and bread of $5 : 1$. Agent j thinks that 1 bread has the same value as 4 apples, i.e. an exchange ratio of $4 : 1$. Take apples as the unit of account. The price of apples is 1, and the price of bread is set at the minimum of the two exchange ratios: $p = \min\{4, 5\} = 4$. Agent i wants to trade in the ratio $5 : 1$ or better, so at the current exchange ratio of $p = 4$ he offers 4 apples in exchange for 1 loaf of bread. Agent j accepts and trades 1 bread for 4 apples. After the exchange both agents are better off than before.

6.4.2 The soup model

The soup model consists of bilateral interactions between randomly paired agents (see Fig. 5). The agents can have the following characteristics:

- Randomly distributed preferences.
- Randomly distributed initial endowments.
- Random pairings:
  - Sequential or parallel exchange.
  - Synchronous or asynchronous exchange.
  - Ex post, a random graph of interactions obtains.

[7] This is the pricing rule being used in Wilhite, 2001.

The soup model is technically a mean field approach. A well-known criticism of the mean field approach is that it is based on the assumption of homogeneity of the agent population. In principle the agents can be heterogeneous, but they are in an infinitely large population, so central limit theorems apply. However, for finite population sizes this does not hold analytically. The mean field approach is connected to an important normative research question: Does the large-type limit hold? That is: *Is it true that if we increase the number of agents in the model we reach similar results as the central limit theorem predictions?* Another question is how the results of large-scale agent-based models scale as we increase the number of agents. Do small- and medium-size populations produce the same aggregate phenomena as large-size populations? For this reason it may be fruitful to consider the random interaction networks, even though it must be conceded that real-world interactions are not completely random.

6.4.3 Wilhite's protocol

Wilhite (2001) uses networks in combination with bilateral exchange mechanisms. The complete network is the baseline case, which he calls the Global Network. In the Global Network every agent can trade with every other agent, in principle. Subsequent restrictions on the trade network limit the trade relationships of the agents to a subset of the population. There is a search-and-trade procedure that proceeds as follows:

- Step 0. Selection step. Select an agent at random, say agent i, who is called the ***search agent***. All subsequent draws from the population are without replacement.
- Step 1. Ranking step. Agent i ranks all the other agents according to their willingness to trade (order of the MRS's). She then selects a potential trading partner on the basis of the best price offered (highest MRS).
- Step 2. Negotiate a price. Agent i and agent j negotiate over the best trading price. They set the trade price according to some price setting rule.
- Step 3. Trading step. Trade continues between the agents i and j in unit increments, until no more utility improvements can be made. Trade stops.
- Step 4. Selection step. A second ***search agent*** is drawn at random from the agent population $A - \{i\}$. Return to step 1.

The process continues until all agents have performed a search-and-trade sequence. This is called one trading round. The process of selecting the search-agents at random can be implemented before the process starts by performing a random permutation of the agent set. Then the search for a potential trade partner and the trading proceed sequentially until all the agents have been a search-agent once. This rule ensures that all the agents have had their opportunity to trade at least once per trading round, although any agent may choose to abstain. During the search process any agent can be selected as a potential trade partner, but trade requires consent. Therefore, in network terminology, link formation requires the consent from both nodes in the network before a link can be formed.
An equilibrium in the bilateral trading process is a rest point of this search-and-trade process. It is reached when no more agents are willing to trade. That is, when all agents choose to abstain when selected as a potential trade partner in step 1 by any search agent. The equilibrium allocation is called Pairwise Optimal, since all potential for pairwise utility improvements have been exhausted. Theoretically it was shown in Feldman (1973) that if all

agents possess a positive amount of some commodity, lets call it money, then a Pairwise Optimal allocation is also a Pareto Optimal allocation. In network terminology, an equilibrium is a rest point of the link formation process when no more nodes want to add links.

7 Learning mechanisms

For a thorough discussion on learning in agent-based models, we refer to the survey in Chapter 6 of the book by Camerer (2003) and to the survey by Brenner (2006) in the ACE Handbook. The following inventory of learning algorithms is based on these two sources.

- Evolutionary algorithms
- Reinforcement learning
- Bush-Mosteller learning
- Roth-Erev learning
- Belief-based learning
- Experience-weighted attraction (EWA) learning
- Melioration learning from own experience
- Imitation learning from group experience
- Social learning by communication

Notation

Let us first illustrate some basic notation commonly used to describe learning algorithms. Given is a set of strategies $\mathbf{S} = (s^1, ..., s^J)$ for N agents $i = 1, ..., N$.[8] A strategy is a mapping from a set of states $x \in \mathbf{X}$ to a set of actions $a \in \mathbf{A}$: $s_i^j : \mathbf{X} \to \mathbf{A}$. The recommended action by strategy s_i^j at time t is given by $a_j(t) = s_i^j(x(t))$, which we sometimes just abbreviate to "action j" for convenience. The agent has a *propensity to use* or an *attraction towards using* a given strategy $s_i^j \in \mathbf{S}$. The attraction to use strategy s_i^j in period t is denoted by $A_i^j(t)$. The reward (or profit, utility, or fitness) for using strategy s_i^j will be denoted by $\pi_i(s_i^j, s_{-i})$. It depends on the collection of strategies $s_{-i} \in \prod_{k \neq i} \mathbf{S}$ that is being used by the other agents. It is important to note that the agents' choice is between the strategies $(s^1, ..., s^J)$ rather than between the actions they prescribe. Since at any given time the selected strategy j prescribes a particular action $a_j(t)$, we may refer to this selection as choosing 'strategy j' or as making the 'choice j' from the choice set, as this comes down to the same. Equating the choice set with the strategy set instead of the action set allows us to have discrete choices in the strategy space, while the action space may be very large.

Summarizing, the notation is as follows:

- There are N actors: $i = 1, ..., N$.
- There are J strategies: $j = 1, ..., J$.

[8] A generalization would be to allow for different strategy sets for different agents: $\mathbf{S}_i = (s_i^1, ..., s_i^J)$.

- s_i^j is the j-th strategy of i's strategy set $\mathbf{S} = (s^1, ..., s^J)$.
- $a_j(t)$ is the recommended action at time t by strategy s_i^j: $s_i^j(x(t)) = a_j(t)$.
- $s_i(t)$ is the strategy that agent i is actually using in period t.
- $s_{-i}(t)$ is the collection of strategies that are actually being used by the other actors in period t.
- $\pi(s_i(t), s_{-i}(t))$ is agent i's payoff from using strategy $s_i(t)$ in period t, while others are using strategies $s_{-i}(t)$.
- $A_i^j(t)$ is agent i's attraction for using strategy s_i^j in period t.

7.1 Evolutionary algorithms

Evolutionary approaches assume an agent is born with an inherent strategy and plays it, usually in random matching with members of a population. Successful strategies increase the agent's relative fitness, giving some relative advantages (frequency of reproduction, length of life, etc.). Evolutionary models generally apply best to animals with genetically heritable strategies or to human cultural evolution (think about genes and Dawkins' memes). Evolutionary mechanisms are well-suited for modelling population dynamics, but less to represent individual learning processes and economic learning behavior. Evolutionary algorithms have been successfully applied in the field of distributed artificial intelligence as optimizing algorithms for machine learning.
Learning mechanisms that fall under the category of Evolutionary Algorithms are: Replicator dynamics, Genetic algorithms, Learning Classifier Systems, and the selection-mutation equation (or Fisher-Eigen equation).

7.2 Reinforcement learning

[Based on: Camerer (2003)]
Reinforcement approaches (also called stimulus-response or rote learning) are one step higher than evolutionary models in the cognitive sophistication that is used by the agents. Choice reinforcement assumes that strategies are *reinforced* by their own previous payoffs, but not necessarily by the payoffs of other strategies. Reinforcement may also *spill over* to other strategies that are similar to the chosen strategy (e.g., neighboring strategies, if strategies are rank ordered). Reinforcement learning is a reasonable model for players with very imperfect reasoning ability or for human players who know absolutely nothing about the forgone or historical payoffs from strategies they did not choose. Another motivation for using a reinforcement learning mechanism comes from the psychology literature which views reinforcement learning as a form of non-conscious learning. No attention or cognition is required by the players if strategies that have performed well in the past are getting reinforced.

A general form of the reinforcement algorithm runs as follows:

0. Initialize the choice propensities/attractions to initial values: $A_i^j(0) = 1$, for $j = 1, ..., J$.
1. Generate the choice probabilities for all actions using the current attractions.
2. Choose an action according to the current choice probability distribution.

3. Update the attractions/propensities for all actions. Depending on the learning mechanisms that is used, this may depend on the reward for the last chosen action, and possibly also on the rewards for non-selected actions.

4. Repeat from step 1.

Learning rules can now be characterized by how the attractions are updated in response to the agent's own experiences, or the direct or indirect observation of the experience of others' strategies.
We start with step 1 of the algorithm, assuming that the attractions have been initialized to some initial values before we compute the choice probabilities.

Step 1. Generating choice probabilities. In order to define an agent's probability to use a certain strategy, the attractions are mapped into predicted *choice probabilities* using some statistical rule. For example, a simple rule would be to use the *linear choice* rule:

$$p_{i,j}(t) = \frac{A_i^j}{\sum_{k=1}^J A_i^k(t)}. \tag{4}$$

A problem with this linear choice rule is that it is too deterministic in some sense. The rule contains no experimentation mechanism, so if certain rules are not being used in the population then the corresponding attractions remain zero, and the choice propensities are zero as well. To ensure sufficient exploration of the rule space, one could simply add a random selection term to the attraction: $A_i^j + \epsilon$.
However, it is currently more common in the literature on social interactions to let the choice probabilities follow the Boltzmann distribution:

$$p_{i,j}(t) = \frac{exp[\beta A_i^j(t)]}{\sum_{k=1}^J exp[\beta A_i^k(t)]}, \text{ for all } i = 1, ..., N. \tag{5}$$

This formulation is called the *discrete choice-* or *multinomial logit* model and can be derived from a random expected utility framework, see McFadden (1973), Diks and van der Weide (2003, p. 4) and Hommes (2006b, p. 1149). The parameter β is often referred to as the *intensity of choice* and is related to the randomness in the strategy selection process. The larger the value of β, the smaller the noise level in the random expected utility, and the larger the probability for an agent to choose the strategy with the highest attraction. For $\beta = \infty$ the random utility term vanishes and the strategy with the highest attraction is chosen. For $\beta = 0$ the random term dominates and all strategies are selected with equal probability. The value of $1/\beta$ can then be interpreted as the propensity of agents to err, if their intention is in fact to select the strategy with the highest attraction.[9]
The discrete choice mechanism represents a general probabilistic framework for strategy selection motivated by results from interacting particle systems in physics, see e.g. Blume (1993) and Föllmer (1974). For use of this framework in models of herding and social

[9] According to the Wikipedia: "The Boltzmann distribution is often expressed in terms of $\beta = 1/kT$ where β is referred to as thermodynamic beta. The term $exp(-\beta E_i)$ or $exp(-E_i/kT)$, which gives the (unnormalised) relative probability of a state, is called the Boltzmann factor and appears often in the study of physics and chemistry." (Source: `http://en.wikipedia.org/wiki/Boltzmann_distribution`). Here E_i stands for the energy of a particle in state i. Since energy is a potential for change, it corresponds to a negative attraction. Hence the attraction terms A_i^j are negative energy terms, and the corresponding factors in our case are given without the minus sign: $exp(\beta A_i^j)$.

interactions, see Brock and Durlauf (2001a,b), and the book on Social Dynamics by Durlauf and Young (2001). For surveys on the use of this framework in models of financial markets, see Hommes (2006a).

Step 2. Choosing an action according to the choice distribution. The choice probabilities generated in step 1 are used to build a cumulative distribution function $F(.)$. To determine an agent's actual choice, a random variable u is drawn from a uniformly random distribution between zero and one. The value of u is then compared to the cumulative distribution function. Action j, $1 \leq j \leq J$, is chosen if $F(j-1) \leq u \leq F(j)$. Action 1 is chosen if $0 < u < F(1)$, and action J is chosen if $F(J-1) < u < 1$.

Step 3. Updating the attractions. A possible form for the updating of the attractions using reinforcement learning is given by the following rule:[10]

$$A_i^j(t) = \phi A_i^j(t-1) + (1-\phi) I(s_i^j, s_i(t)) \pi(s_i(t), s_{-i}(t)), \tag{6}$$

where ϕ is a discount factor which depreciates previous attraction, and $I(x,y)$ is an indicator function which equals 1 when $x = y$ and 0 otherwise:

$$I(s_i^j, s_i(t)) = \begin{cases} 1, \text{ if } s_i^j = s_i(t), \text{ hence strategy } j \text{ is actually used by } i \text{ in period } t. \\ 0, \text{ if } s_i^j \neq s_i(t), \text{ hence strategy } j \text{ is not used by } i \text{ in period } t. \end{cases} \tag{7}$$

In words, the attraction of strategy j in the current period t, $A_i^j(t)$, is equal to:

- the depreciated attraction from the previous period, $A_i^j(t-1)$, if strategy j was not used by agent i in period t.
- idem, plus the payoff from using strategy j, if agent i actually *did* use strategy j in period t.

The updating equations for the attraction of choice j and choice k are:

$$A_i^j(t) \quad = \phi A_i^j(t-1) + (1-\phi)\pi(s_i(t), s_{-i}(t)), \tag{8}$$
$$A_i^k(t) \quad = \phi A_i^k(t-1), \text{ for } k \neq j. \tag{9}$$

Several learning mechanisms, such as Bush-Mosteller learning, Roth-Erev learning, Belief-based learning and EWA learning can all use the same algorithm above to update the choice probabilities and select a strategy. The only difference is in the computation of the attractions.

7.3 Bush-Mosteller learning

[Based on: Brenner (2006)]
The Bush-Mosteller learning mechanism assumes that the learning process is a Markov process. The frequency distribution $\mathbf{p}(t)$ is independent of choices or outcomes in the past, it only depends on outcomes in the current period. The Bush-Mosteller mechanism does distinguish between positive and negative outcomes, which corresponds to findings in the psychological literature. The updating equations for the attractions are as follows.

[10]This requires that the attractions A_i^j and the payoffs π are measured in the same units.

If strategy s_i^j is selected and the outcome is positive, $\pi_i(t) \equiv \pi(s_i^j, s_{-i}) \geq 0$, then:

$$A_i^j(t+1) = A_i^j(t) + v(\pi_i(t))(1 - A_i^j(t)), \text{ for choice } j \tag{10}$$
$$A_i^k(t+1) = A_i^k(t) - v(\pi_i(t))A_i^k(t), \text{ for choice } k \neq j. \tag{11}$$

It is clear that the positive outcome for choice j should increase the attraction of j and decrease the attraction of k. The function v is a monotonically increasing function of π on the unit-interval: $v(0) = 0$, $v' > 0$, $0 \leq v(\pi) \leq 1$ with $v(\pi) \rightarrow 1$ as $\pi_i \rightarrow +\infty$.

If strategy s_i^j is selected and the outcome is negative, $\pi_i(t) \equiv \pi(s_i^j, s_{-i}) < 0$, then:

$$A_i^j(t+1) = A_i^j(t) - v(-\pi_i(t)) \cdot A_i^j(t), \text{ for choice } j \tag{12}$$
$$A_i^k(t+1) = A_i^k(t) + v(-\pi_i(t))\frac{A_i^k(t) \cdot A_i^j(t)}{1 - A_i^j(t)}, \text{ for choice } k \neq j. \tag{13}$$

A negative outcome for choice j leads to a decrease in the attraction of j and to an increase in the attraction of choice k, proportional to the attractions for both strategies.

Bush-Mosteller learning could be used in combination with Kahneman and Tversky's Prospect Theory, in which utility (performance, fitness, reward) is defined in terms of gains and losses of wealth rather than in absolute levels. Kahneman and Tversky propose a value function $v(.)$ of the following form:

$$v(\pi) = \begin{cases} \pi^\alpha, \text{ if } \pi \geq 0, \\ -\lambda(-\pi^\beta), \text{ if } \pi < 0, \end{cases} \tag{14}$$

where π is the gain ($\pi \geq 0$) or loss ($\pi < 0$) and λ is a coefficient of loss aversion. Kahneman and Tversky estimated α and β to be 0.88 and λ to be 2.25.

7.4 Roth-Erev learning

[Based on: Marks (2006)]
The general Roth-Erev model of reinforcement learning takes two forms: without experimentation and with experimentation. The learning mechanism without experimentation is an example of non-conscious learning. The learning mechanism with experimentation is an example of cognitive learning, since experimentation requires that agents pay attention to the performance of the non-used strategies, and this requires a conscious effort.
The variables and parameters are as in Table 2. The equations for updating the attractions for the selected strategy s_i^j and the non-selected strategies s_i^k, $k \neq j$ are as follows:

$$A_i^j(t) = (1-\phi)A_i^j(t-1) + \phi\pi(s_i^j, s_{-i})(1-\epsilon), \tag{15}$$
$$A_i^k(t) = (1-\phi)A_i^k(t-1) + \phi\pi(s_i^j, s_{-i})\frac{\epsilon}{J-1}. \tag{16}$$

The attraction $A_i^j(t-1)$ from the previous period is down-weighted by the recency parameter ϕ. When $\phi = 0$ all weight is put on the previous attraction to select j, and no weight is put on the current reward $\pi(s_i^j, s_{-i})$ for the most recent choice j. When $\phi = 1$, the previous attraction does not matter and only the reward for the most recently chosen action matters. This means that agents switch immediately to the action that yields the highest reward in the previous period.

The updating equations for the attractions can be parameterized by introducing a function $E_k(j, t, \epsilon, J)$:

$$\begin{aligned} A_i^j(t) &= (1-\phi)A_i^j(t-1) + \phi E_k(j, t, \epsilon, J), \text{ for all } j = 1, ..., J \\ E_k(j, t, \epsilon, J) &= \begin{cases} \pi(s_i^j, s_{-i})(1-\epsilon) & \text{if } k = j \text{ (for choice } j), \\ \pi(s_i^j, s_{-i})\frac{\epsilon}{J-1} & \text{for other choices.} \end{cases} \end{aligned} \tag{17}$$

The term E_k depends on the selected strategy j and its payoff $\pi(s_i^j, s_{-i})$ in period t, the experimentation parameter ϵ and the total number of choices J. The dependence can be explained in two parts.

First, there is the influence of the most recent payoff for choice j on the propensity for choosing j again. If experimentation is small (ϵ is small) the reward is directly added to the propensity to chose j. In the attraction for choosing j, the current payoff of j plays a large role: $\phi\pi(s_i^j, s_{-i})(1-\epsilon)$. If experimentation is large (ϵ close to 1) then the reward for j is unimportant in choosing the next action, since large experimentation approximates random choice.

Second, there is the influence of the reward from strategy j on the propensity to chose a different strategy $k \neq j$. In this case, the propensity to switch strategies is linearly related to the experimentation parameter ϵ. In general, the probability to switch is inversely proportional to the total number of choices available. In the attraction for choosing a different strategy k, the current payoff of j plays only a small role: $\phi\pi(s_i^j, s_{-i})\frac{\epsilon}{J-1}$.

Table 2: Variables and parameters in the Roth-Erev learning model with experimentation.

Variables		Parameters	
j	most recent choice	$A_i^k(0)$	initial propensities
A_i^j	propensity to select strategy j	ϵ	experimentation parameter
A_i^k	propensity to select strategy k	ϕ	recency parameter
$\pi(s_i^j, s_{-i})$	reward for strategy j	J	number of choices

An extension to the above formulation of Roth-Erev learning was introduced by Nicolaisen et al. (2001) who modify the function $E_k(j, t, \epsilon, J)$ as follows:

$$A_i^j(t) = (1-\phi)A_i^j(t-1) + \phi E_k(j, t, \epsilon, J) \tag{18}$$

$$E_k(j, t, \epsilon, J) = \begin{cases} \pi(s_i^j, s_{-i})(1-\epsilon) & \text{if } k = j, \\ A_i^k(t-1)\frac{\epsilon}{J-1} & \text{otherwise.} \end{cases} \tag{19}$$

Instead of using the reward from the selected strategy j, now the previous attraction from the non-selected strategy k is used, in combination with the experimentation parameter ϵ. This results in the following updating equations:

$$A_i^j(t) = (1-\phi)A_i^j(t-1) + \phi\pi(s_i^j, s_{-i})(1-\epsilon), \tag{20}$$

$$A_i^k(t) = (1-\phi)A_i^k(t-1) + \phi A_i^k(t-1)\frac{\epsilon}{J-1}. \tag{21}$$

This modifies the recency parameter for the non-selected strategies to a lower value: $\phi^\star = \phi - \frac{\epsilon}{J-1}$ (see Marks, 2006, p. 1365). Hence, in the computation of the attractions a recently used strategy gets more weight than a non-selected strategy by lowering its recency parameter somewhat.

7.5 Belief-based learning

[Based on: Camerer (2003)]
Belief-based learning models assume agents update their beliefs about what others will do based on history, and use those beliefs to determine which strategies are best. It therefore ignores information about the own choices in the past. Two classic examples of belief-based learning are *fictitious play* and *Cournot best-response.*
In fictitious play, players keep track of the relative frequency with which another player has played each strategy in the past. These relative frequencies are beliefs about what that player will do in the upcoming period. Players then calculate expected payoffs for each strategy given these beliefs, and choose strategies with higher expected payoffs more frequently. Fictitious play counts all previous observations equally.
Cournot best-response dynamics assumes that the strategy played most recently by other agents will be played again. Weighing distant experiences less than recent ones gives a hybrid form called *weighted fictitious play* (Cheung and Friedman, 1997).
An example of updating beliefs in weighted fictitious play is the following:

$$E_i(s_{-i}^j)(t) = \frac{I(s_{-i}^j, s_{-i}(t)) + \sum_{k=1}^{t-1} \phi_i^k I(s_{-i}^j, s_{-i}(t-k))}{1 + \sum_{k=1}^{t-1} \phi_i^k} \quad t = 1, 2, ... \tag{22}$$

Adding the weight ϕ is sensible because standard fictitious play ignores the fact that another player may have made various choices in the past. Cournot best-response dynamics errs in the opposite direction by taking into account only what happened in the previous period.
Weighted fictitious play is a sensible compromise. When $\phi = 1$, weighted fictitious play reduces to origina1 fictitious play; when $\phi = 0$, it is Cournot best-response.
It can be shown that the attractions update as:

$$A_i^j(t) = \frac{\phi N(t-1) A_i^j(t-1) + \pi_i(s_i(t), s_{-i}(t))}{\phi N(t-1) + 1}, \tag{23}$$

where $N(t)$ weighs experience in the sense specified in the following section on EWA learning.

7.6 Experience-weighted attraction (EWA) learning

[Based on: Camerer (2003)]
Reinforcement learning assumes that agents ignore information about forgone payoffs (payoffs from strategies that were not actually used), while belief-based learning assumes that agents ignore information about their own past choices, relying only on their beliefs about the choices of others. But agents in the real world seem to use both types of information when it is available.
EWA learning (Camerer and Ho, 1999; Camerer et al., 2002; Camerer, 2003) is a family of learning rules of which reinforcement learning and belief-based learning are special cases. It incorporates two learning effects, namely the *law of actual effect* and the *law of simulated effect.*
The law of actual effect refers to the fact that agents learn from information about the own past choices. Selected strategies that were successful in the past will have a higher probability to be selected in the future. This law is also at the core of reinforcement learning mechanisms such as Roth and Erev (1995) (see also Marks, 2006; Pouget, 2007).
The law of simulated effect refers to the notion that agents learn from information about others' choices in the past. The agent observes (i.e. simulates) the payoffs from non-selected strategies and reinforces the successful ones. This law is at the core of belief-based learning.

EWA learning therefore is a hybrid form of learning, combining the law of actual effect and the law of simulated effect. This hybrid model consists of two variables: the attractions $A_i^j(t)$ and an experience weighing parameter $N(t)$, which are both updated after every period of experience. The model adds a key feature to reinforcement and belief-based learning models which is the weight given by players to forgone payoffs from unchosen strategies, denoted by the parameter δ. See Table 3 for a reference to the EWA learning parameters.

Table 3: Variables and parameters in the EWA learning model.

Variable	Description
$N(t)$	experience/number of previous observations
$A^j(t)$	attraction to select strategy j
$\pi^j(t)$	reward from using strategy j at time t
Param.	Description
ρ	'memory' parameter: depreciation rate of previous observations $N(t-1)$
ϕ	'change' parameter: depreciation rate of previous attraction $A^j(t-1)$
δ	'imagination' parameter: law of simulated effect, weight on forgone payoff $\pi^j(t-1)$
κ	'lock-in' parameter, $\rho = (1-\kappa)\phi$: $k = 0 \Leftrightarrow \rho = \phi$

The experience weight starts at an initial value $N(0)$ and is updated according to

$$N(t) = \rho N(t-1) + 1, \tag{24}$$

where ρ is a depreciation rate that measures the fractional impact of previous experiences, compared to one new observation in the current period. If $\rho = 0$ all previous experiences are fully discounted and has no impact on the current strategy selection: there is no memory effect and $N(t) = 1 \forall t$. If $\rho = 1$ then there is a strong memory effect and past observations/experiences are fully taken into account. The experience weight in this case reduces to a counter of the number of observations: $N(t) = N(t-1) + 1$.
Attractions are initiated at $A_i^j(0)$ and are updated according to:

$$A_i^j(t) = \frac{\phi N(t-1) A_i^j(t-1) + [\delta + (1-\delta) I(s_i^j, s_i(t))] \pi_i(s_i^j, s_{-i}(t))}{N(t)} \, , \tag{25}$$

where the factor ϕ is the discount factor that depreciates the previous attraction, $N(t-1)$ weighs the experience, and δ is the weight given to the forgone payoffs from the unchosen strategies.
If $\phi = 0$ this means that previous attractions are completely discounted, and the attraction equals the performance measure: $A_i^j = \pi^j(t)$. If $\phi = 1$ this means that previous attractions are not completely discounted but taken into account in the computation of new attractions, which now are a weighted sum of experience (the performance) and attraction (hence the term Experience-Weighted Attraction learning).
The value of $\delta \in [0, 1]$ can take two extreme values. When $\delta = 0$ only the law of actual effect is used, while when $\delta = 1$ this implies that both the law of actual effect and the law of simulated effect are used. Any value in between discounts the forgone payoffs from the unchosen strategies with δ. When $\delta = 0$, the attractions simply represent the cumulated past payoffs from the used strategies only. When $\delta = 0, \rho = 0$, the past experiences do not matter (note that $N(t) = 1$ in every period), so there is no memory effect and EWA learning reduces to reinforcement learning, compare (6).

When $\delta = 1$ and $\rho = \phi$, the attractions of EWA learning in (25) reduce to those given for weighted fictitious play, compare (23).[11]
In a later version of the EWA learning framework, the parameter ρ has been replaced by $(1-\kappa)\phi$, see Eqn. 2 in Camerer et al., 2002, p. 6. This implies that when $\kappa = 0$, the same case as $\rho = \phi$ is obtained. A cube with all possible parameter configurations (δ, ϕ, κ) appears in Camerer et al., 2002, p. 42, Fig. 2. Special cases then appear as edges or corners of this EWA learning cube. In Table 4 we provide a summary of the properties of EWA learning for these eight extreme cases.
The formula in (25) can now be split up into two parts, relating to the law of actual effect and the law of simulated effect respectively:

$$\begin{cases} A_i^j(t) = \frac{\phi N(t-1) A_i^j(t-1) + \pi_i(s_i^j, s_{-i}(t))}{N(t)}, & s_i^j = s_i(t), \\ A_i^j(t) = \frac{\phi N(t-1) A_i^j(t-1) + \delta\pi_i(s_i^j, s_{-i}(t))}{N(t)}, & s_i^j \neq s_i(t). \end{cases} \tag{26}$$

In the first line, the law of actual effect as measured in (25) by the term $[\delta + (1-\delta)I(s_i^j, s_i(t))]\pi_i(s_i^j, s_{-i}(t))$ equals $\pi_i(s_i^j, s_{-i}(t))$ since $I(s_i^j, s_i(t)) = 1$. Hence, the reward from the chosen strategy j is fully taken into account in the attraction for strategy j. In the second line, the law of simulated effect is measured by the term $\delta\pi_i(s_i^j, s_{-i}(t))$, since it weighs the foregone payoff of strategy j, $\pi_i(s_i^j, s_{-i}(t))$, even though strategy j was not actually chosen by agent i. Nonetheless, it is given some weight δ in the updating of the attraction $A_i^j(t)$ for strategy j.
A general form of the EWA learning algorithm runs as follows (see also the algorithm for reinforcement learning at the beginning of this chapter):

0. Initialize the choice propensities/attractions to initial values: $A^j(0)$, for $j = 1, ..., J$.
1. Generate the choice probabilities for all actions using the current attractions.
2. Choose an action according to the current choice probability distribution.
3. Update the attractions for all actions.
4. Repeat from step 1.

[11] In human subject experiments, estimates of δ are generally around .50, ϕ around .8 to 1, and ρ varies from 0 to ϕ (see Camerer and Ho, 1999).

‘memory’	‘imagination’	‘change’	‘lock-in’	description
ρ	δ	ϕ	$\kappa = 1 - \rho/\phi$	
0	0	0	0	no memory, law of actual effect only, attraction equals performance.
0	1	0	0	no memory, law of actual and simulated effect, attraction equals performance.
0	0	1	1	no memory, law of actual effect only, attraction discounted.
0	1	1	1	no memory, law of actual and simulated effect, attraction discounted.
1	0	0	–	memory, law of actual effect only, attraction equals performance.
1	1	0	–	memory, law of actual and simulated effect, attraction equals performance.
1	0	1	0	memory, law of actual effect only, attraction discounted.
1	1	1	0	memory, law of actual and simulated effect, attraction discounted.

Table 4: EWA parameter settings. Eight possible configurations of EWA learning parameters. Note that $\rho \equiv (1-\kappa)\phi$, hence $\kappa := 1-\rho/\phi$. A cube with all possible parameter configurations (δ, ϕ, κ) appears in Camerer et al., 2002, p. 42, Fig. 2, with special cases denoted as edges or corners. The eight cases in the table correspond to the cases in this EWA learning cube. Extreme cases are: $\rho = 0$: no memory, $\rho = 1$: memory. $\delta = 0$: reinforcement learning (law of actual effect), $\delta = 1$: belief-based learning (law of simulated effect). $\phi = 0$: attraction equals performance, $\phi = 1$: attraction weighs performance. Rule 1: reinforcement learning, rule 2: fictitious play, rule 8: Cournot best-response; weighted fictitious play.

7.7 Melioration learning from own experience

Melioration learning is based solely on the individual's own experiences. As agents observe the outcomes of their actions over time, experiences are collected. Experiences from different situations are lumped together if the circumstances are perceived to be sufficiently similar. Individuals then retain a memory of past experiences that can be used when a similar situation occurs in the future.
Melioration learning is an adjustment process that is a special case of fictituous play (see Brenner, 2006, p. 910). For each action a, the utility or payoff $u(a,t)$ is calculated on the basis of a finite memory of past choices. Then, again for each action a, the average payoff $\bar{u}(a,t)$ is calculated, which denotes the average of the utilities obtained from selecting action a in the finite memory. The relative frequencies $p(a,t)$ with which an action a gets selected over any other action b depends on the relative difference between the average payoffs obtained over the finite memory horizon:

$$\frac{dp(a,t)}{dt} = v(\bar{u}(a,t) - \bar{u}(b,t)),. \tag{27}$$

with $v(0) = 0$, $v' > 0$ (monotonically increasing). A possible extension is to use an exponentially weighted average of past payoffs:

$$\bar{u}(a,t) = \frac{1-\beta}{1-\beta^{t-1}} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} u(a,\tau) \cdot I(a(\tau) = a), \tag{28}$$

where $I(a(\tau) = a)$ is an indicator function for whether the action a was in fact selected in time-period τ. The discount rate β discounts the memory of past actions, such that more recent actions have a higher weight in the calculation of the weighted average payoff.

7.8 Imitation learning from group experience

Imitation depends on the performance of non-selected strategies versus the performance of the current strategy. Imitation is often a good economizing heuristic because agents only need to repeat the observed strategy, rather than having to form beliefs and evaluate all available strategies (see Schlag, 1999).
In models of learning by imitation we should distinguish between the number of other individuals an agent can observe:

- only individuals in the local neighborhood are observed.
- only a single other individual is observed, drawn at random from the population.
- a randomly picked subpopulation of the entire population is observed.
- the entire population is observed.

In the psychological literature imitation learning is studied under the label of observational learning (Brenner, 2006, p. 912). The agent's information set consists of the collected experiences from the others, that can be either directly or indirectly observed. Since melioration learning also refers to the collection of experiences it is also a form of observational learning. The same models can then be used to model both types of learning, since both refer to the process of collecting experiences over time.

The same formulation can be used as before (see melioration learning above), but now the average payoff to individual i includes the experiences from all N individuals in the population $(1, ..., N)$:

$$\bar{u}_i(a,t) = \frac{1-\beta}{1-\beta^{t-1}} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} \left[\sum_{j=1}^{N} \sigma(i,j) \cdot u_j(a,\tau) \cdot I(a_j(\tau)=a) \right],$$
$$\text{with } \sum \sigma(i,j) = 1, \tag{29}$$

where $u_j(a,\tau)$ is the payoff obtained by individual j at time τ by selecting action $a_j(\tau)$, and $\sigma(i,j)$ is the weight attached by individual i to individual j's experience.

7.9 Social learning by communication

If the learning model also includes communication with others then the agent's information set can be further extended to include all those experiences that have been *gathered* by others. The only two modelling parameters that remain to be set are then:

- How much of others' experiences can be obtained?
- How much weight is given to the other agents' communicated experience?

Learning by communication can be modelled by adding a weight specifying whether the other's 'experience' is its own experience, or whether it is an observation by that individual.
The way to model this is to differentiate between the weights given to first-hand experience and to second-hand experience, and so on. The individuals own experience gets the highest weight, then the experiences it obtains from observing others' behavior, then the experiences that are communicated by the others:

$$\bar{u}_i(a,t) = \frac{1-\beta}{1-\beta^{t-1}} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} \left[\begin{array}{l} \sigma(i,i) \cdot u_i(a,\tau) \cdot I(a_i(\tau)=a) \\ + \sum_{j=1}^{N} \sigma(i,j) \cdot u_j(a,\tau) \cdot I(a_j(\tau)=a) \\ + \sum_{j=1}^{N} \sum_{k=1}^{N} \sigma(i,j) \cdot \sigma(j,k) \cdot u_k(a,\tau) \cdot I(a_k(\tau)=a) \end{array} \right], \tag{30}$$

with $\sigma(i,i) > \sigma(i,j) > \sigma(j,k)$ and $\sigma(\cdot,\cdot) \in [0,1]$. The third weight $\sigma(j,k)$ is the weight that individual i attaches to the communication by individual j of its observations of individual k. Of course, the experience of individual j should not have the same weight as its communications, since these reflect second-hand information for individual i.
Note however that in the summations the own payoffs of i are being counted double, which reflects the idea that if individual i receives a positive communication from individual j about its own actions then the actions gets reinforced.

Summarizing, (Brenner, 2006, p. 909-913) provides a general model of learning that combines the following features:

- Experience collection: there is some positive weight attached to the own payoffs from actions selected in the past, for some finite memory horizon.
- Imitation: the past experiences of others can be obtained and are taken into account in the decision-making.
- Memory: The memory of past payoffs is weighted over time, using a hyperbolic form with an exponentially weighted memory decay.

- Communication: There is experimentation and communication between the agents with respect to the non-selected strategies.

7.10 Conscious and non-conscious learning mechanisms

[Based on: Brenner (2006)]
Conscious reflection on behavior applies to situations that are unfamiliar and hence require attention. Such reflection is restricted by the time available for deliberation and contemplation. Therefore, in order to prevent the cognitive system from being overloaded with trivialities, attention needs to be economised. Only those situations that truly demand a conscious deliberation effort will receive attention, all others are dealt with in a non-cognitive way, i.e. by using previously established behavioral routines.
As long as the non-conscious, routine behavior yields satisfactory outcomes (with respect to to some aspiration level), the agent continues to use its routines. A dissatisfaction with the outcomes of the routine behavior motivates a change in behavior. This triggers a conscious learning phase in which renewed attention is given to those situations for which the routines have failed to give satisfactory outcomes. In the conscious learning phase a new behavioral routine is established. This routine is then again continuously applied in the non-conscious learning phase as long as satisfying outcomes are obtained. This learning procedure thus consists of a conscious learning phase and a non-conscious learning phase, described below.

Conscious learning phase:

- Continue to use cognitive learning rules as long as the behavior remains unsatisfactory.
- Switch to the non-conscious learning phase if satisfying behavior has been learned.

The termination criterion for the conscious learning phase is based on a satisficing method. The learned behavior is deemed satisfactory only if a certain aspiration level has been reached. Otherwise the agent continues to use the cognitive learning mechanism to adapt its behavioral routines.

Non-conscious learning phase:

- Continue to use the behavioral routines as long as satisfactory outcomes are obtained.
- Switch to the conscious learning phase if outcomes prove unsatisfactory.

The termination criterion for the non-conscious learning phase is also conditioned on the aspiration level. As soon as the outcome drops below the aspiration level it is deemed unsatisfactory and this triggers the switch to the conscious learning phase.

Classification of conscious and non-conscious learning mechanisms

Non-conscious learning mechanisms: [See also: Brenner (2006), p. 901 for a similar categorization.]

- Evolutionary algorithms: random mutation, cross-over and selection as the basic driving forces for strategy selection.
  - Replicator dynamics
  - Genetic algorithms
  - Selection-mutation equation (Fisher-Eigen)
- Reinforcement learning: based on past performance, successful behavior gets reinforced and unsuccessful behavior is diminished.
  - Bush-Mosteller model
  - Roth-Erev model without experimentation and forgetting

Conscious learning mechanisms:

- Routine-based learning: there is a direct connection from the individual's observations and own experiences to their behavior.
  - Melioration learning
  - Imitation learning
  - Roth-Erev model with experimentation and forgetting
- Belief-based learning: based on the individual's observations and experiences (of self and others), beliefs are formed on the strategies used by others.
  - Fictitious play
  - Cournot best-response
  - Weighted fictitious play
  - Experience-weighted attraction (EWA)
- More complicated belief-based learning: an individual's mental model of the environment consists of more complex schemata.
  - Genetic programming
  - Classifier systems
  - Neural networks
  - Rule learning
  - Baysian learning
  - Least-squares learning
  - Stochastic belief learning

Pseudocode for conscious and non-conscious learning mechanisms

Algorithm 7.1 LEARNING ALGORITHM

Is the situation well-known and repeated?

- YES: Start the non-conscious learning phase
- NO: Start the conscious learning phase

```
Start of conscious learning phase:
repeat conscious learning mechanism
    collect experiences of self
    observe experiences of others
    communicate experience to/from others
    test outcome against aspiration level
until aspiration level is reached and satisfying behavior is learned
then switch to non-conscious learning phase

Start of non-conscious learning phase:
repeat non-conscious learning mechanism
    collect experiences of self
    observe experiences of others
    test outcome against aspiration level
until aspiration level is violated and unsatisfactory outcome is obtained
then switch to conscious learning phase
```

8 Classification of learning mechanisms

A classification of the learning mechanisms entails that we need to make explicit the relationships between the learning models and the situational characteristics of each market. The link between a particular learning model and a particular market then runs through the agent's learning competencies. The first step is to determine what are the informational requirements for successful learning to take place on each market. Secondly, we need to determine which learning mechanisms satisfy these informational requirements. Then we can link the learning mechanisms to the markets based on the requirements such that agents can use the learning mechanism in an adequate fashion.

There are three ways in which an individual can obtain information: through personal experiences, through the observation of others, or through communication with other agents. Broadly speaking, this corresponds to three levels of learning: individual learning from own experience, group learning from group experience, and social learning from population experience. We can categorize this into three classes of information sets:

- Self observation: the agent observes only its own experiences.
- Local observation: the agent can observe the experiences of other agents, but is restricted to a local neighborhood.
- Global observation: the agent can observe some global information on the aggregate level (global statistics).

Given the above subdivision, we will provide a categorization of the agents' information set on each market. This facilitates the selection of the appropriate learning mechanism for each market, since it simply becomes a matter of matching the characteristics of the appropriate learning mechanism to the informational requirements of the agents on each market.
Below we discuss the following categorizations:

- Categorization of learning models: For each learning mechanism, what are the informational requirements? Does the mechanism use memory, the experiences of others, and is there local or global observation of others' strategies?
- Selection of learning mechanisms for each market: What are the informational requirements for the agents in each market context?

8.1 Categorization of learning models

In order to categorize the learning algorithms, we can classify them according to their inputs (experience and observations).

Type of inputs:

- Memory [+/-]: The previous experience has/does not have a weight in the learning mechanism.
- LAE [+/-]: Law of actual effect, the performance of the own strategy is observed/not observed.
- LSE [+/-]: Law of simulated effect, the performance of other strategies can be observed/can not be observed.
- Observation [S/L/G]: Self-, local- or global observation, agents can observe only the own experiences, part of other strategies, or can observe all of other strategies.
- Experimentation [+/-]: The learning mechanism includes/does not include experimentation with non-selected rules.

In the section above several learning mechanisms have been identified. These learning mechanisms are commonly used in Behavioral Economics and agent-based modelling. They fit quite naturally into the above classification scheme. Table 5 classifies the learning mechanisms according to their respective informational requirements.
In the table we distinguish between four classes of learning mechanisms: evolutionary learning, reinforcement learning, routine-based learning and belief-based learning.
The first distinction is that evolutionary and reinforcement learning mechanisms do not have a memory of past events, while routine-based and belief-based mechanisms do have memory, with the possible exception of Cournot-best response (only relies on the most recently played strategy).
Within each class we make a second distinction, which is whether the mechanism uses just the own experience to update the attractions (the law of actual effect), or whether in addition the observations of others' experiences are also taken into account (the law of simulated effect).
Almost all mechanisms use the law of actual effect, with the exception of imitation learning, fictitious play and Cournot best-response, which do not use the information about the own strategy performance but only use the performance of the other, non-selected strategies. These

mechanisms are purely responsive to opponents' behavior, instead of reinforcing the own behavior.
A third distinction is that between local and global observations. If the mechanism makes use of the experience of others, this may consist of local observation of the performance of local neighbours, or observing the performance of all strategies. This defines the scope of the learning mechanism.
A fourth and final distinction is whether the mechanism uses experimentation to switch between strategies. Within the class of evolutionary mechanisms, both learning classifier systems and genetic algorithms use similar evolutionary operators to modify the rule system (selection, reproduction, cross-over, mutation and election). In the class of belief-based mechanisms it is only EWA learning that includes an experimentation mechanism. In general, the EWA learning mechanism is the most versatile, since it encompasses several learning mechanisms; reinforcement learning, belief-based learning, Cournot best-response, and weighted fictitious play are all incorporated as special cases.

8.2 Selection of learning mechanisms for each market

What do agents need to know in order to learn on a particular market? This depends on the decisions they need to make. So for each market we can specify what information is required for successful learning to take place on that particular market. Table 6 provides a categorization of the agents' information set on each market, and the appropriate learning mechanism corresponding to that information set. Below we give a description for each market.

Labor market The interactions on the labor market between employers and employees is such that there is no strategic interaction. Households applying for a job do not form beliefs about the hiring policies of their prospective future employer. Similarly the firm does not take into account the job searching strategy of its applicants. They do not have information on the other side of the market.
Concerning the information that is observable, a firm can only access its own previous hiring history, not the experience of other firms. A firm can observe the current state of a job seeking agent, but not its employment history. Furthermore, all observations are local. A firm only knows the employment state of the local households that have applied for a job.
Also, the job applicants do not observe the history of other job applicants. A worker who is employed by a firm does not observe the experiences of any of the other workers inside or outside of the firm. The only incentive to quit is the worker's own job-satisfaction level. To some extent this is a restrictive modelling assumption, since we cannot include between-worker communication. The forecasting horizons on the labor market are limited to one-step ahead forecasts, without a memory of past experiences.

Selected learning mechanisms: Bush-Mosteller, Roth-Erev learning without experimentation.

Investment goods market The market for investment goods is a centralized market. Therefore all information on the investment goods market is global information. This implies that an investment goods producer knows the prices of all of its competitors. A buyer of investment goods also has this global information. Furthermore, both sides of the market need to make forecasts about their own future actions and future payoffs, in order to make sound investment decisions. To that end they use a long-term planning horizon and this requires a memory of past experiences. However they do not need to take into account the future actions of others and do not need to make forecasts about the others' payoffs. Therefore, the information is global but agents use only their own experiences to select the next action and possibly future actions.

Selected learning mechanisms: Conscious learning. Roth-Erev with experimentation, EWA learning, Melioration learning.

Consumer goods market Due to the high degree of habitual and routine-driven behavior on the consumer goods market, and due to the relative unimportance that is usually attached to consumption decisions, this would lead to a recommendation of using a non-conscious learning mechanism. However, if unknown situations are encountered (such as the introduction of a new consumer product) it would be more appropriate to use a conscious learning mechanism.
In general, buyers on the consumer goods market do not observe the behavior of other buyers (or their performance). Hence only the own experience matters. Furthermore, consumers do not attach great significance to the past. Habitual consumption patterns are hard to break even if a household's income is drastically reduced. Most consumer behavior is quite stubborn in this respect and learning could be modelled using a non-conscious learning mechanism that uses only the most recently observed own performance, without a memory of previous experience. Buyers usually do not produce forecast beyond their own next period payoffs, so the forecasting horizon is extremely short.
The recommendation for habitual consumption is thus to use a form of reinforcement learning, such as Bush-Mosteller, or Roth-Erev without experimentation.

Word-of-mouth effects are also important on the consumer market. These could be modelled using a group learning or social learning mechanism. The recommendation for non-habitual consumption is then to use a conscious learning mechanism such as Roth-Erev with experimentation, Experience-weighted attraction, Imitation learning, or a Communication learning mechanism.

On the seller's side of the market the learning is much more cognitive. Market-based research and advertizing campaigns are decisions that are made deliberately and consciously. Information that is used consists of the own experiences (both past and present) with a memory. A firm cannot obtain the experience of another firm. Only local information is obtained by firms about the current actions and current performances of their competitors. Furthermore, firms can only obtain local information on the buyers' current experiences (for example using consumer surveys), and only on consumers that are in their local neighborhood. The forecasting horizon is limited to one-step ahead predictions of the own next payoffs.

Selected learning mechanisms:

Buyers: Non-conscious learning. Reinforcement mechanisms, Bush-Mosteller, Roth-Erev without experimentation.

Buyers: Conscious learning. Roth-Erev with experimentation, Experience-weighted attraction, Imitation learning, Communication learning.

Sellers: Conscious learning. EWA learning, Learning Classifier Systems.

Credit market Due to the relative importance that can be attached to the decisions on the credit market, the decision-making process has a high degree of rationality and therefore a conscious learning mechanism is recommended.
On the credit market there is an informational asymmetry, since the credit applicants (the households or firms) have less information than the credit providers (the banks). The applicants have a memory, but use only their own experience. They have only local observations and do not experiment. The banks have memory, use their own experience and have global information. Since only the own experience is used, this limits the selection of learning mechanisms.

Selected learning mechanisms: Melioration learning.

Asset market Financial traders can obtain information on the performance of different trading strategies by observing the payoffs of other traders or by paying for investment advice. They can also observe all strategies that are being used (this is a strong assumption). The information therefore consists of a global observation of the experiences of others. Again, due to the importance of the decisions on the asset market this leads to the recommendation of using a conscious learning mechanism.
In general, financial traders can be characterized as having short forecasting horizons. However, the forecasts pertain to the future performance of the own strategy as well as to the future performance of other strategies, which makes imitation learning and EWA learning particular meaningful in this market context.

Selected learning mechanisms: Conscious learning. Belief-based learning, Imitation learning, EWA learning, Learning Classifier Systems.

Learning mechanism	Type of inputs required					Description
	memory	LAE	LSE	obs.	exp.	
Evolutionary algorithms						Non-cognitive, non-conscious learning mechanisms.
- Replicator dynamics	-	+	-	S	-	
- Learning Classifier Systems	-	+	+	G	+	A set of classifier rules (condition-action pairs).
- Genetic algorithm	-	+	+	G	+	Population-level adaptation. Selection is based on all performances.
Reinforcement learning						Non-conscious, stimulus-response learning.
- Bush-Mosteller	-	+	-	S	-	Uses only own experiences.
- Roth-Erev w/o experimentation	-	+	-	S	-	Neglects the foregone payoffs of non-selected strategies.
Routine-based learning						Quasi-conscious learning mechanisms.
- Melioration learning	+	+	-	S	-	Uses only own experience (finite memory).
- Imitation learning	+	-	+	LG	-	Uses own experience and observations of others (finite memory).
- Roth-Erev with experimentation	+	+	-	L	+	Uses own experience and observations, weighs past performance.
Belief-based learning						Conscious learning, beliefs about others' through experience.
- Fictitious play	+	-	+	G	-	Tracks relative frequencies of past play.
- Cournot best-response	-	-	+	G	-	Assumes most recently played strategy will be repeated.
- Weighted fictitious play	+	-	+	G	-	Hybrid form between fictitious play and Cournot best-response.
- Experience-weighted attraction	(+)	+	+	G	(+)	Uses own experience and observations of others.

Table 5: Categorization of the learning mechanisms according to their type and information input/output. LAE=Law of actual effect (payoff of used strategy), LSE=Law of simulated effect (payoff of non-used strategy), obs.: S=self, L=local, G=global, exp.=experimentation.

Market	Type of inputs obtainable					Recommended learning mechanism
	memory	LAE	LSE	obs.	exp.	
Labor market	-	+	-	S	-	Bush-Mosteller
						Roth-Erev w/o experimentation
Investment goods market	+	+	-	G	-	Roth-Erev with experimentation
						EWA learning
						Melioration learning
Consumer goods market						Bush-Mosteller learning
Buyers	-	+	-	S	-	Roth-Erev w/o experimentation
						Imitation learning
						Roth-Erev with experimentation
Sellers	+	+	-	L	-	EWA learning
						Roth-Erev with experimentation
Credit market	+	+	-	S	-	Melioration learning
Asset market	(+)	(+)	+	LG	-	Imitation learning
						Roth-Erev with experimentation
						EWA learning
						Learning Classifier Systems

Table 6: Categorization of the agents' informational requirements for several market contexts, and the recommended learning mechanisms. LAE=Law of actual effect (payoff of used strategy), LSE=Law of simulated effect (payoff of non-used strategy), obs.: S=self, L=local, G=global, exp.=experimentation.

9 Simulation output, data storage and user interfaces

We are still considering what statistics are to be gathered during run-time, what data on individual agents to store, and what kind of probing tools are to be used at run-time to follow the agents.

9.1 Data storage

From the very start is seems sensible to think about what data should be stored from every simulation run. Although highly desirable, storing absolutely everything may not be the best course of action. The restrictions on what to store may serve as a disciplinary device for the modeller.
The most extreme case is to keep every action of every agent and every interaction between all the agents that has taken place, including all the possible plans of agents that have not resulted in any action. In short, this entails storing a full history for every run and this means that it will never be necessary to reproduce any run with exactly the same initial conditions, since it can be retrieved from storage. This requires clear estimates of what will be the storage requirements for the simulator as a whole.
An aspect to think about is the distinction between data available at run-time and data that is stored for later data analysis. Suppose one run yields 1 GB of data, which is not a lot if there are 10^6 agents, each yielding 1 MB of data. Suppose further that there are 100 runs for 10 scenarios each. That would yield $1,000$GB of data to store and analyse afterwards. With these kinds of estimates it becomes relatively easy to run into serious data management issues.

Example. Data estimates. Consider an estimate for an economic agent using approx. $O(10^5)$ bytes of memory to store its 'lifetime' history. For the EURACE economy that contains $O(10^6)$ agents this would imply:

$$O(10^5) \text{ bytes/agent } * O(10^6) \text{ agents } = O(10^{11}) \text{ bytes of RAM} = 100\text{GB of RAM}. \tag{31}$$

This is the amount of RAM that would be required to keep all agents in memory for one run. But for testing the simulator we can test on the order of $O(10^4)$ agents and be happy with 1GB of RAM presumably. Ten such runs yield 1TB of data, and 200 runs yield 20TB. In comparison, the U.S. Library of Congress has claimed it contains approximately 20 terabytes of text.

9.2 Graphical output and user interface

Here we need a list of what type of graphics we would need: time series plots, phase plots, vertical bar charts, horizontal bar charts, how to visualize distributions, etc. The economists could start by making a 'wishlist':

- It would be nice to have a window with the real geographical map of the EU, with the $100 \times 100km$ grid, and the NUTS level 1 regions.
- It would be nice to have a color-coding scheme that indicates the levels of a particular macroeconomic variable of interest.
- It would be nice to have the possibility of clicking on a country, and that this would produce an in-depth view of the country's state variables (key macroeconomic indicators).

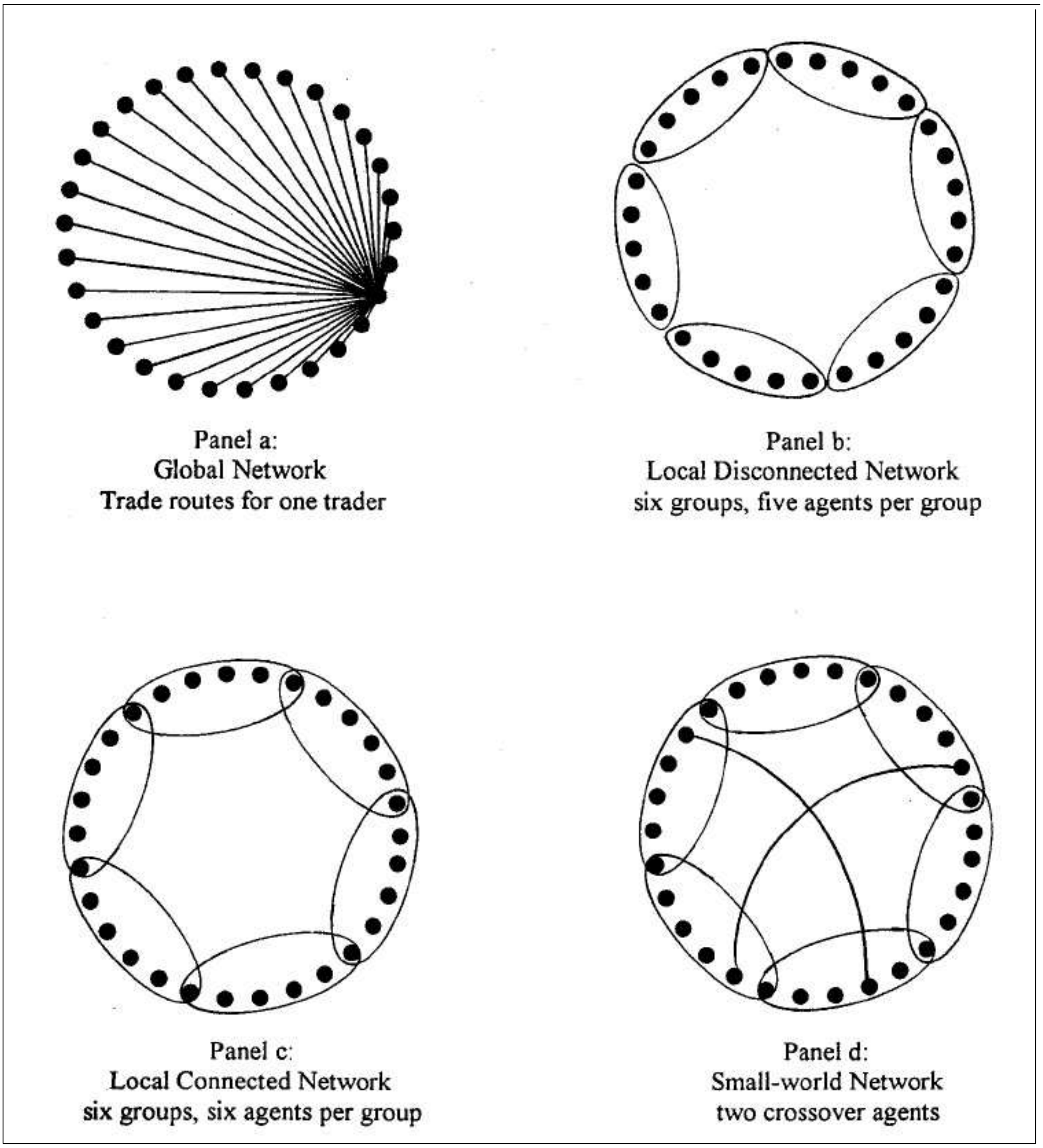

Figure 1: Network structure. Source: Wilhite (2001).

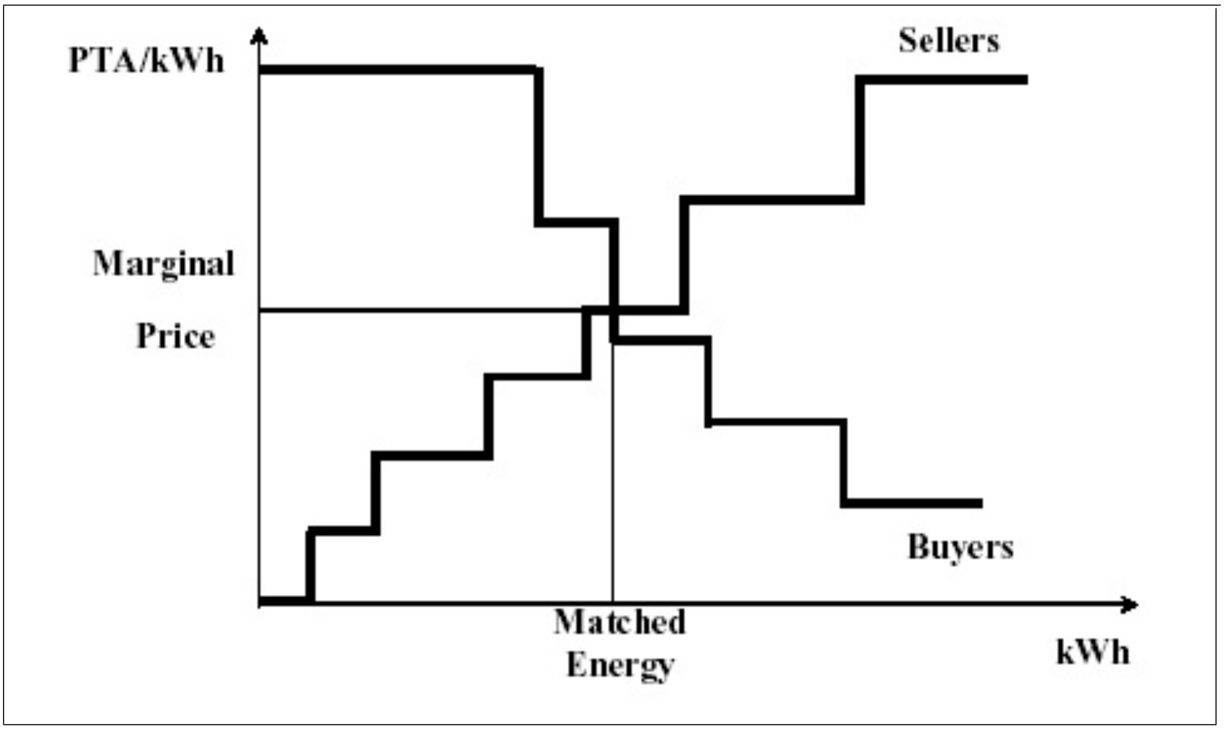

Figure 2: Illustration of the clearinghouse mechanism with aggregate demand and supply schedules. Source: Gonzalez and Basagoiti (1999).

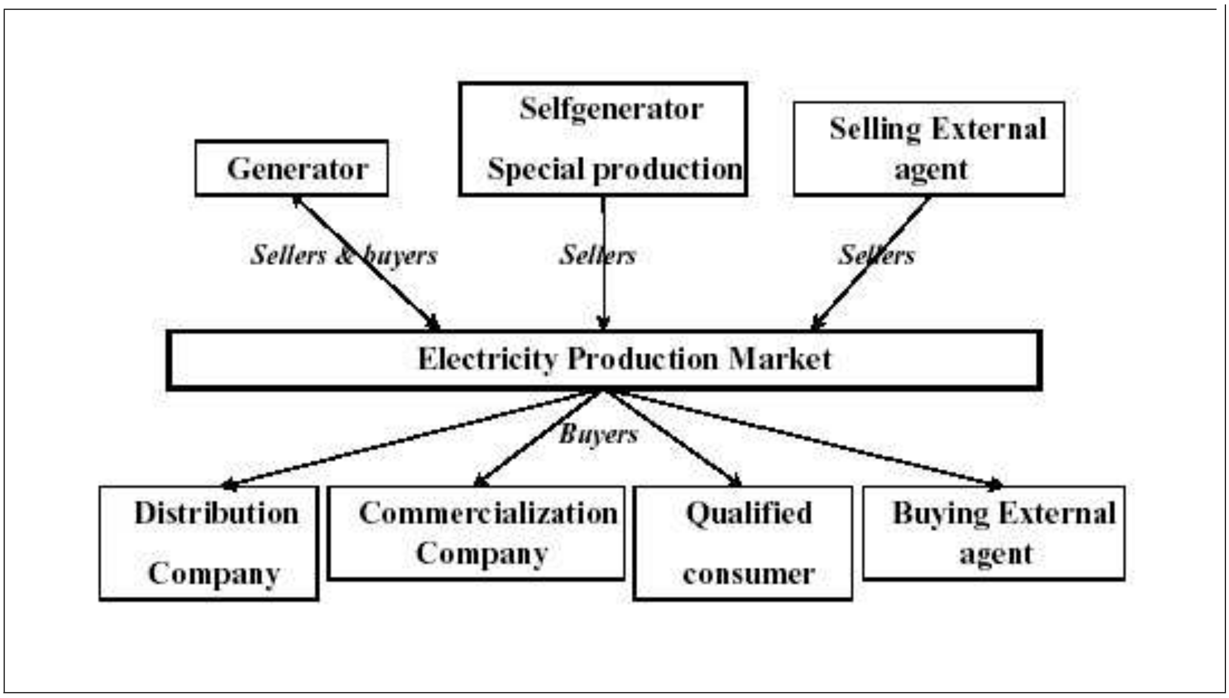

Figure 3: Market organisation in the Spanish Energy Market. Source: Gonzalez and Basagoiti (1999).

Figure 4: A random process model of the continuous double auction. Stored limit orders are shown stacked along the price axis, with sell orders (supply) stacked above the axis at higher prices and buy orders (demand) stacked below the axis at lower prices. New sell limit orders are visualized as randomly falling down, and new buy orders as randomly 'falling up'. New sell orders can be placed anywhere above the best buying price, and new buy orders anywhere below the best selling price. Limit orders can be removed spontaneously (e.g. because the agent changes her mind or the order expires) or they can be removed by market orders of the opposite type. This can result in changes in the best prices, which in turn alters the boundaries of the order placement process. The horizontal gap between the best buying price and the best selling price is called the 'market spread'. Orders may be entered inside the spread without being executed, thereby decreasing the size of the spread. When a market order is entered against the best buying or best selling price, the size of the spread can be increased. Figure taken from Farmer et al. (2003), p. 10.

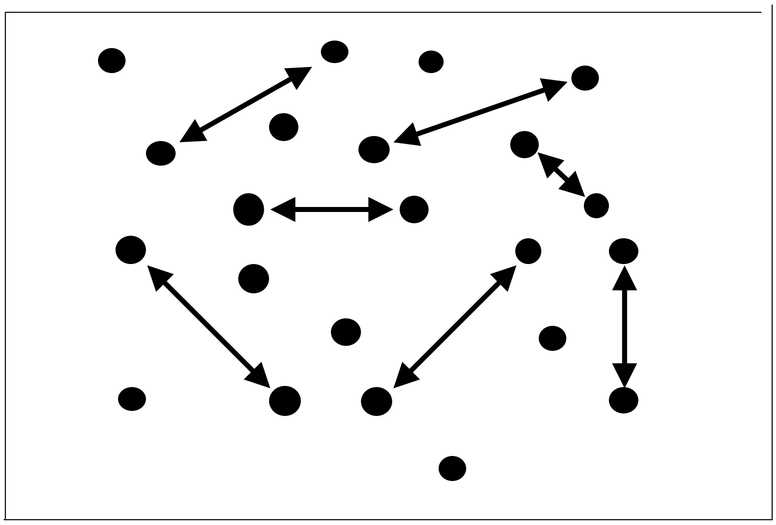

Figure 5: Random agent pairing in the mean field approach: the ‘soup’ model.

References

Albin, P., Foley, D. K., 1992. Decentralized, dispersed exchange without an auctioneer: A simulation study. Journal of Economic Behavior and Organisation 18, 27 – 51.

Amihud, Y., Mendelson, H., Spring 1991. Trading mechanisms and value-discovery: Cross-national evidence and policy implications. Vol. 34 of Carnegie-Rochester Conference Series on Public Policy. pp. 105 – 130.

Axtell, R. L., July 2000. Effects of interaction topology and activation regime in several multi-agent systems. Vol. 1979 of Lecture notes in computer science. Springer, Berlin, Boston MA, pp. 33 – 48.

Axtell, R. L., June 2005. The complexity of exchange. The Economic Journal 115, 193 – 210.

Bajari, P., Hortaçsu, A., 2003. The winner's curse, reserve prices, and endogenous entry: empirical insights from ebay auctions. RAND Journal of Economics 34 (2), 329 – 355.

Barabasi, A.-L., Albert, R., 1999. Emergence of scaling in random networks. Science 286.

Bell, A. M., 1997. Bilateral trading on a network: convergence and optimality results. Working paper No. 97-W08, Vanderbilt University.

Binmore, K., Klemperer, P., 2002. The biggest auction ever: the sale of the british 3G telecom licences. Economic Journal.

Blume, L., 1993. The statistical mechanics of strategic interaction. Games and Economic Behavior 5, 387 – 424.

Brenner, T., 2006. Agent learning representation: Advice on modelling economic learning. In: Tesfatsion, L., Judd, K. L. (Eds.), Handbook of Computational Economics, Volume 2: Agent-Based Computational Economics. Elsevier, North-Holland, pp. 895 – 947.

Brock, W. A., Durlauf, S. N., 2001a. Discrete choice with social interactions. The Review of Economic Studies 68 (2), 235 – 260.

Brock, W. A., Durlauf, S. N., 2001b. Interactions-based Models. Vol. 5 of Handbook of Econometrics. Ch. 54, pp. 3297 – 3380.

Camerer, C., 2003. Behavioral Game Theory. Princeton University Press.

Camerer, C., Ho, T., 1999. Experience-weighted attraction learning in normal-form games. Econometrica 67, 827 – 74.

Camerer, C., Ho, T., Chong, J.-K., 2002. Sophisticated ewa learning and strategic teaching in repeated games. Journal of Economic Theory 104(1), 137–188, `http://faculty.haas.berkeley.edu/hoteck/PAPERS/paper2.pdf`.

Cespa, G., 2004. A comparison of stock market mechanisms. RAND Journal of Eonomics 35 (4), 803 – 823.

Cheung, Y., Friedman, D., 1997. Individual learning in normal form games: Some laboratory results. Games and Economic Behavior 19, 46 – 76.

Cyert, March, 1963/92. A Behavioral Theory of The Firm. Prentice Hall., Englewood Cliffs, NJ.

de Solla Price, D. J., 1965. Networks of scientific papers. Science 149, 510–515.

de Solla Price, D. J., 1976. A general theory of bibliometric and other cumulative advantage processes. Journal of American Society of Information Sciences 27, 292–306.

Dessalles, J. L., Phan, D., 2004. Emergence in multi-agent systems: cognitive hierarchy, detection, and complexity reductions. Working Paper ELICCIR.

Diks, C. G., van der Weide, R., 2003. Heterogeneity as a natural source of randomness. CeNDEF Working paper 03-05, University of Amsterdam, August 14, 2003.

Durlauf, S. N., Young, H. P., 2001. Social Dynamics. MIT Press.

Farmer, J. D., Patelli, P., Zovko, I., 2003. The predictive power of zero intelligence models in financial markets. Los Alamos National Labratory Condensed Matter.

Feldman, A., 1973. Bilateral trading processes, pairwise optimality, and pareto optimality. Review of Economic Studies 40 (4), 463–473.

Föllmer, H., 1974. Random economies with many interacting agents. Journal of Mathematical Economics 1, 51 – 62.

Gonzalez, J. J., Basagoiti, P., July 1999. Spanish power exchange market and information system design concepts, and operating experience. In: Power Industry Computer Applications. Proceedings of the 21st 1999 IEEE International Conference. Santa Clara, CA, USA, pp. 245 – 252, URL: http://www.econ.iastate.edu/tesfatsi/sp1epmp.pdf.

Guerci, E., Ivaldi, S., Raberto, M., Cincotti, S., 2005. Agent-based simulation of power exchange with heterogeneous production companies. Working paper University of Genoa, URL: http://econpapers.repec.org/paper/scescecf5/334.htm.

Holcombe, M., Coakley, S., Smallwood, R., October 2006. A general framework for agent-based modelling of complex systems. EURACE Working paper WP1.1, Department of Computer Science, University of Sheffield.

Hommes, C. H., 2006a. Heterogeneous Agent Models in Economics and Finance. Elsevier, North-Holland, handbooks in economics series 23, pp. 1109–1186.

Hommes, C. H., 2006b. Interacting Agents in Finance. New Palgrave Dictionary of Economics (2nd ed.). Palgrave Macmillan.

van der Hoog, S., Deissenberg, C., 2007. Modelling specifications for EURACE. EURACE Report D2.1, GREQAM, Université de la Méditerranée.

Klemperer, P., 2000. Auction Theory: A Guide to the Literature. The Economic Theory of Auctions. Edward Elgar, Ch. 1.

Klemperer, P., 2002a. How (not) to run auctions: the european 3g telecom auctions. European Economic Review.

Klemperer, P., 2002b. Some observations on the british and german 3g telecom auctions. Economic Studies.

LeBaron, B., Arthur, W. B., Palmer, R., 1999. Time series properties of an artificial stock market. Journal of Economic Dynamics and Control 23, 1487 – 1516.

Marks, R., 2006. Market design using agent-based models. Handbook of Computational Economics, Volume 2: Agent-Based Computational Economics. Elsevier, Elsevier:North-Holland, handbooks in economics series 27, pp. 1339 – 1380.

McFadden, D., 1973. Conditional logit analysis of qualitative choice behavior. Frontiers in Econometrics. Academic Press, NY, New York, pp. 105 – 142.

Newman, M. E. J., 2003. The structure and function of complex networks. SIAM Review 45 (167).

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

Pouget, S., 2007. Adaptive traders and the design of financial markets. Journal of Finance (forthcoming).

Pryor, R., Marozas, D., Allen, M., Paananen, O., Hiebert-Dodd, K., Reinert, R., 1998. Modeling requirements for simulating the effects of extreme acts of terrorism: A white paper. Report SAND98-2289, SANDIA National Laboratories.

Roth, A. E., Erev, I., 1995. Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term. Games and Economic Behavior 8, 164–212.

Sargent, T. J., 1993. Bounded Rationality in Macroeconomics. Oxford University Press, USA.

Schlag, K., 1999. Which one should I imitate? Journal of Mathematical Economics 31, 493 – 522.

Simon, H. A., 1955. On a class of skew distribution functions. Biometrika 42, 425–440.

Spanish Market Authority, April 2001. Electricity market activity rules (non binding unofficial translation). URL: `http://www.omel.es/es/pdfs/EM-Rules.pdf`.

Swedish Competition Authority, November 1996. Deregulation of the swedish electricity market. Report 1996 – 3, Swedish Competition Authority.

Wilhite, A., 2001. Bilateral trade and 'small-world' networks. Computational economics 18, 49 – 64.

Wolfstetter, E., 1999. Topics in Microeconomics, Industrial Organization, Auctions, and Incentives, 1st Edition. Cambridge University Press.

Modelling Specifications for EURACE

Responsible authors:
Sander van der Hoog
Christophe Deissenberg
(GREQAM, Université de la Méditerranée)

Contributions from:
Simon Coakley (Sheffield)
Herbert Dawid (Bielefeld)
Michael Neugart (Bielefeld)
Michele Marchesi (Cagliari)
Marco Raberto (Genoa)
Andrea Teglio (Genoa)

Workpackage: WP2.1 : Agent-based computational economics
Date: October 9, 2007
Contributing units: Bielefeld, Genoa, Cagliari, Sheffield
Responsible unit: GREQAM, Université de la Méditerranée
Distribution level: Public domain
Discussed at meeting(s): Ancona Meeting, 27 October 2006; Nice Meeting, 26-28 January 2007.

Contents

Abstract

This document contains the modelling specifications for the EURACE simulator.

Acknowledgements

This work was supported by the European Union through its 6^{th} Framework Programme, FET-IST 'Complex Systems'. Funding for the STREP research project 'EURACE' under contract no. 035086, is gratefully acknowledged. The following persons have contributed to this paper. Simon Coakley contributed to Section 2, Michele Marchesi contributed Section 4.1, Marco Raberto and Andrea Teglio contributed Section 4.2, and Herbert Dawid, Michael Neugart and collegues contributed Section 4.3. Discussions on the topics in this paper have been held at various stages of development, during two EURACE meetings: the first held in Ancona, 27 October 2006 and the second held in Nice, 26-28 January 2007. The usual disclaimer applies.

1 Introduction

In the first setup of the modelling guidelines we listed two key documents, the MRD called 'Modelling Requirements for EURACE' (see van der Hoog and Deissenberg, 2007) and the MSD, called 'Modelling Specifications for EURACE'.

The purpose of the MSD (this document) is to provide high-resolution descriptions of the items to be used in the EURACE simulator. Whereas in the MRD the topics listed were all of a very general nature, the MSD will be much more focussed on the issues at hand. The descriptions should be at such a high level of detail that it yields sufficient information for the computer engineer to implement the model element. For example, for the agent types the MSD will not only list all allowable actions for a particular agent type, but it will also need to specify the processes underlying these actions, and under what conditions a particular method is being activated.
The high-resolution descriptions of the MSD should satisfy the criterion of Dynamical Completeness (see Tesfatsion, 2006). Ideally, the MSD should also contain for every model component a Level of Effort (LOE), providing an estimate of how long it will take to implement a particular model element.

Naming conventions

To smooth the model building process, it would be advantageous to keep an up-to-date listing of all the naming conventions that are being used for variables and functions. Especially with regard to the interface between different markets it seems important to have such a list. A proposal for the naming conventions is made on the following EURACE Wiki page:

- `http://www.eurace.org/Wiki/index.php/Naming_Conventions`

2 Agent architecture: Agents as X-machines

In order to specify blueprints for our economic agents, we adhere to the computational paradigm of automata. In particular, we model the agents as ***X-machines***, which are automata with an internal memory. X-machines use transition functions to model the process, communication control and data flow, through one iteration of a model. The behaviour of the X-machine is defined by the order of the transition functions. See Holcombe et al. (2006) and Coakley and Kiran (2007) for further details on the implementation.

2.1 X-machines

An X-machine consists of an internal memory, an internal state, and inputs and outputs. It takes as inputs its current internal state q_1, the internal memory m_1 and (optional) input-messages $t_1 \in \Sigma$. The output consists of a new internal state q_2, new internal memory values m_2 and (optional) output-messages $s_2 \in \Gamma$.
The message space specifies the agent's ***communication relation*** R with other agents. It contains all the messages that agents are able to read and/or write. In addition, it is also possible to exclude certain agents from reading or sending certain types of messages.
Formally, Communicating X-machines are defined as follows:

Definition 2.1 *A stream X-machine is an 8-tuple*

$$X = (\Sigma, \Gamma, Q, M, \Phi, F, q_0, m_0) \tag{1}$$

where:

- Σ *and* Γ *are the input and output alphabets respectively.*
- Q *is the finite set of states.*
- M *is the (possibly) infinite set called memory.*
- Φ *is the type of the machine X; it is a set of partial functions* ϕ *that map an input and a memory state to an output and possibly a different memory state,* $\phi : \Sigma \times M \to \Gamma \times M$*.*
- F *is the state transition function,* $F : Q \times M \times \Sigma \to Q \times M \times \Gamma$*, which given the current state, current memory values and (optional) input message determines the next state, new memory values and (optional) output messages.*
- q_0 *and* m_0 *are the initial state and initial memory respectively.*

2.2 X-agents

An ***X-agent*** is an agent defined as an X-machine. It is a blueprint that can be used to create (or 'instantiate') an agent. The computational structure of an economic X-agent consists of:

- Internal state, $q \in Q$: representing the agent's current condition.
- Internal memory, $m \in M$: describing the agent's current information set.
- Input and output messages, Σ and Γ: defining the agent's messages.
- Partial functions $\phi \in \Phi$: describing the agent's communications relations with other agents.

- Transition functions, $f \in F$: describing how the agent's internal state and memory are updated.

Next, in order to refine our definitions of economic X-agents, we need to define for every agent class the following items:

1. The markets on which the agent can be active.
2. The decisions that the agent has to make on each market.
3. The allowable actions that the agent can perform on each market. These can be either:
   - independent actions: only affecting the own internal state and internal memory;
   - interactions: affecting the internal states and internal memory of other agents.
4. The messages that an agent can send to other agents in the environment, and to institutions such as markets.

Finally, for every allowable action we need to specify how the action results from an explicit deliberation process or from an implicit routine. This requires more detailed specifications of the learning algorithms that have been proposed in the EURACE Modelling Requirements, see van der Hoog and Deissenberg (2007).

2.3 Contexts-Roles-Activities

Contexts In general, markets do not act. They do not have agency, since they do not have intentions and cannot be said to perform actions. However, a convenient way to view a market is to see it as providing some *context* for agents to act *in*. If there are certain market mechanisms that define how a particular market functions, then these mechanisms will have to be 'embodied' into an X-agent as well. This is the only way such mechanisms can be incorporated into an X-agent model, since the market as a whole remains only the *context* in which these mechanisms are interacting with the agents. The input and output of market mechanisms are modelled through messages, similar to how active agents communicate.

Roles Agents always act within a certain context, but can have different roles in different contexts, and this will have to be reflected in the way we structure the X-agent definitions. We propose to use the following three-level hierarchy:

- Define the list of agents.
- For each agent, define the contexts in which this agent is supposed to act, and its respective role in each context.
- For each role, define the functions that this role should perform, given the context.

Activities Activities, decisions and tasks are all implemented as functions. This defines the functionality of each role. Suppose we have already specified a list of decisions for each agent type. Each item of that list could now be implemented as a set of functions. Finally, all variables and parameters of the conceptual model need to be implemented as internal memory variables of X-agents, since global variables are absent from the framework.

Table 1: Terminology table. Correspondence between terms in the conceptual model and the implemented model.

Conceptual model	X-agent implementation
agents	X-agents
roles	none (roles are not represented)
activities, decisions, tasks	a set of functions
information signals	messages
variables	internal memory variables
parameters	parameters

Having such a hierarchy will allow us to separate the functions of a single agent into several subclasses that are relevant for each distinct market context, without breaking the possible function dependencies that may exist between functions of different agent roles. All the functions of an agent can have dependencies on every other function of the same agent, irrespective of the roles the agent is playing. Fig. 1 shows examples of such agent-role hierarchies for different types of agents.

Another advantage of using such a hierarchy is that we can use a similar hierarchical structure for the messages. All messages belonging to a certain context can be collected into a subclass of messages. Since messages do not belong to an agent but are defined outside of the agent scope, the message dependencies of the functions of all agent types that are active in the same context are clearly delineated per context. For example, all agents that are active on the same market have a common subset of messages.

Table 1 is a correspondence table, giving a summary of items in the conceptual model and the corresponding item in the implemented X-agent model.

The distinction between different roles only exists within the conceptual model, not in the implemented model, since it does not make sense to implement each role as a separate X-agent. This is so because there can be a function dependency between the functions of different roles within the same agent. If we would then model each role as a separate X-agent we would have to define messages passing between roles, in order to transmit the output values of a certain function that are needed as input values to another function (since function dependencies are only defined with respect to the own set of functions). This is rather cumbersome and indeed unnecessary, so the notion of 'roles' only exists in the conceptual model and does not have a counterpart in the FLAME implementation.

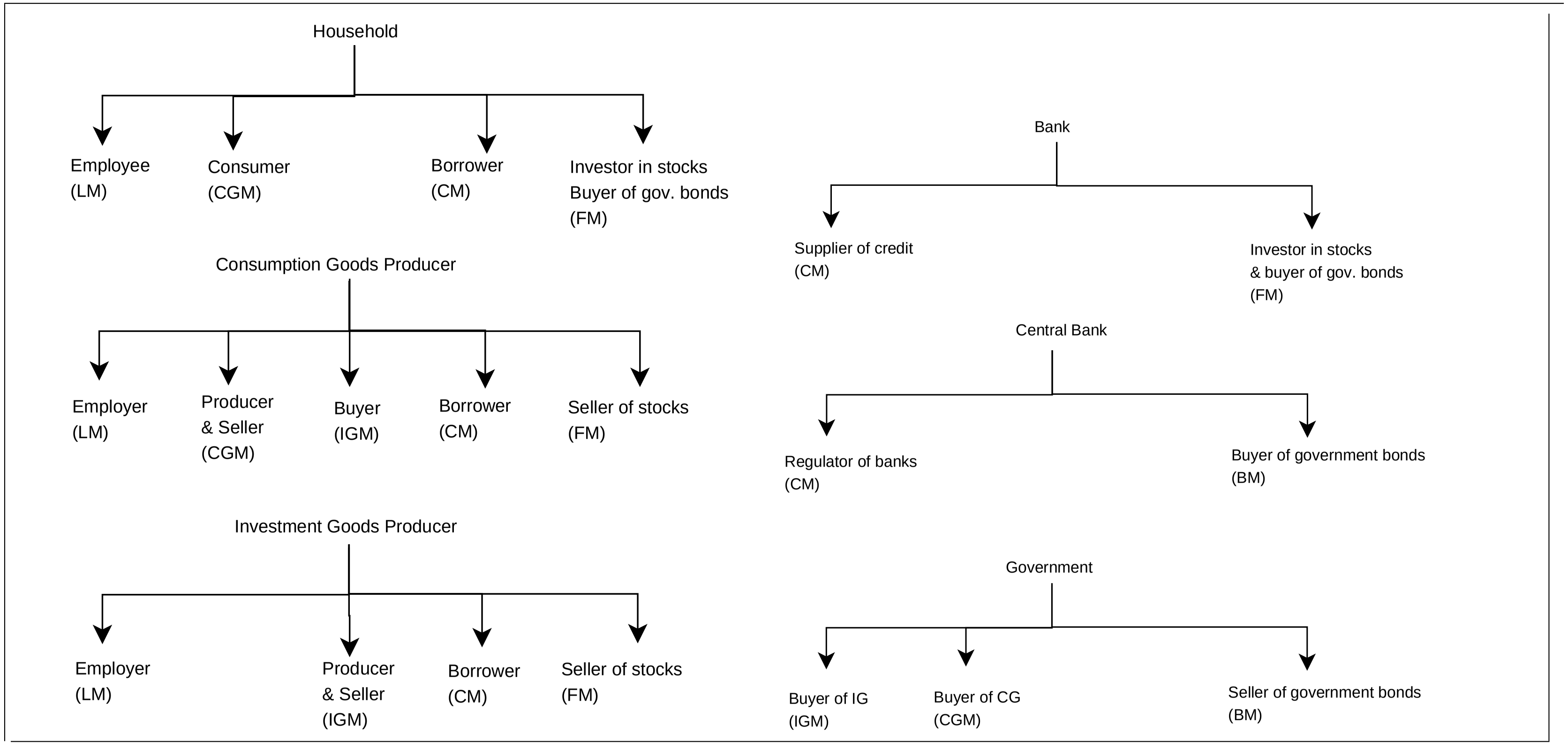

Figure 1: Diagrams of agent-role hierarchies highlighting the different market contexts in which the agents are active.

3 Categorizations

This section provides categorizations of the economic sphere into a number of different classifiers. We use geographical regions, economic sectors, and the type of market as different ways to partition the economy.

3.1 Categorization of economic regions

The target system is the European Union (EU-27), see Figure 3. Thus, we refrain from modelling the entire European Economic Area (EEA), which also contains the European Free Trade Association (EFTA: Iceland, Norway, Switzerland and Lichtenstein).
For the geographical scope of the EURACE framework we propose to use the so called NUTS Regions, which is the statistical classification scheme that is being used by Eurostat. The NUTS classification is a 3-level hierarchy dividing the national territory of each EU Member State into economic regions.[1] The definition of a NUTS region is according to a minimum and a maximum threshold for the average population size in the NUTS region, see Table 2.
The 3 NUTS levels are further subdivided into two Local Administrative Units (LAU level 1, formerly NUTS level 4, and LAU level 2, formerly NUTS level 5). Using the NUTS and LAU levels, the EU-25 consisted of 1214 units at NUTS level 3, and $112,119$ units at LAU level 2 (data from 2003, see Table 3). For the EU-27 we should add the $28+42$ NUTS level 3 regions of Bulgaria and Roumania.

Complete tables and maps of the NUTS regions can be found here:

Basic NUTS information:
`http://ec.europa.eu/comm/eurostat/ramon/nuts/basicnuts_regions_en.html`

Number of units in each NUTS level (Data of 2003):
`http://ec.europa.eu/comm/eurostat/ramon/nuts/introannex_regions_en.html`

Maps and Tables of NUTS level 1 and 2:
`http://ec.europa.eu/comm/eurostat/ramon/nuts/overview_maps_en.cfm`

Excel files with LAUs for each member state:
`http://ec.europa.eu/comm/eurostat/ramon/nuts/lau_en.html`

3.1.1 Geographical scope

For the geographical representation, we propose to use a grid with a grid size of 100×100 km, matching the approximate size of regions at NUTS level 2.
In Figure 4 we show an overlay of the NUTS regions with the 100×100 km grid. To match up the grid with the real geographical area of the European Union, see Table 3 and 4 for some basic data.
Some considerations for linking the NUTS regions to the grid system are the following:

- At which NUTS level/grid size is there a good correspondence?
- In any case, we should have that $\#cells \geq \#regions$. Otherwise there are unrepresented regions and the grid does not provide a good representation of the target system.

[1]See `http://en.wikipedia.org/wiki/Nomenclature_of_Territorial_Units_for_Statistics`.

- Ideally, this procedure will result in a mapping of the geographical distances between different NUTS regions to geographical distances in the grid system.

- The numbers in the tables can also be useful for a possible partitioning strategy for the load-balancing in the parallelized framework. For instance, one could think about distributing the X-agents according to the real demographical distribution of the population in the EU (currently estimated at $492,215,000$, IMF 2007). This directly links the real-world geographical distribution to the distribution of agents across the nodes in the computing cluster. Compare for example the total number of NUTS 3 regions (1284) to the total number of nodes in a cluster. If the cluster contains 128 nodes then it seems logical to have 10 NUTS level 3 regions per node.

3.2 Categorization of economic sectors

3.2.1 Economic structure

For the general economic structure of the EURACE Model we refer to Figure 5. Figure 5 represents the general framework for EURACE. In Section 3.3 the flows are discussed in more detail.
The next sections provide a description of the real and financial part of the EURACE artificial economy.

3.2.2 Economic sectors - Real part

The real part of the EURACE economy (production, distribution and consumption) could consist of the following sectors:

- Sector 1: Investment goods production.

- Sector 2: Consumption goods production.

3.2.3 Economic sectors - Financial part

- Banking sector (business credits and private loans).

- Financial sector (financial markets, i.e., stock markets, government bond markets, other financial assets).

- Government sector (Government, European Central Bank).

Different agent classes correspond to each of these sectors. Below we provide more detail on each agent class. A flow chart of the general framework is provided in Fig. 5.

3.3 Categorization of agent classes

The EURACE economy may be populated by the following classes of agents:

- Households.

- Investment good producers (firms in sector 1).

- Consumption good producers (firms in sector 2).

- Bankers and financial intermediaries.
- Government.
- Central Bank (ECB).
- Rest of the World (ROW, multiple blocks).

Households The households receive income from working, receive private loans from the banking sector and they receive household transfers, unemployment and Social Security benefits from the government. Households hold asset portfolios that pay out dividends, and they have a private savings account with a bank that pays out interest. The earnings (income plus asset accumulation earnings) are spend on the consumption goods market.

Investment good producers The investment good producers use labour and energy as input factors to produce investment goods (technologies). The investment goods are sold to the consumption good producers.

Consumption good producers Consumption goods are produced with labour and physical capital as input factors. The physical capital is purchased on the market for investment goods using financial capital (this includes retained earnings, debt, equity issues). The financial capital is obtained on the asset market by the emission of equity (shares) and corporate bonds. The business loans are obtained from the banking sector through the credit market. The firm pays dividends on its outstanding shares, coupons on outstanding bonds and interest on outstanding debt. It also has to make some debt repayments.

Credit sector Inputs to the credit or banking sector come from the households in the form of private savings. The banks pay out an interest rate on the saving accounts. Outputs are to the households in the form of private loans (i.e., car loans, mortgages), and to the firms in the form of business credits. The banks sell credit on the credit market, and buy and sell assets (stocks and bonds) on the asset markets. The banks can apply for reserves with the central bank, and have to pay a discount rate for this. We abstract from banking sector employment.

Financial sector We consider the following classes of financial instruments:[2]

- A loan is a primary debt instrument that is negotiated between a borrower and a lender and is tailored to the borrower and lender's needs.
- A bond is a primary debt security. Government bonds are issued to finance the current budget deficit.
- A corporate bond is simply a loan, but in the form of a primary debt security that is issued by a corporation in standardized units. The firm issuing the corporate bond is basically a borrower, the bond holder is the lender, and the coupon of the bond is the interest being paid by the firm to the bond holder. The corporate bond represents ownership in the corporation.

[2] Source: `http://en.wikipedia.org/wiki/Debt_security`. We are excluding some important derivative instruments, such as forwards, futures, options, swaps and CMO's. See also `http://www.drfurfero.com/books/2309book/ch01f.html`.

- A share (or stock) is a secondary security that is issued by a corporation in standardized units and represents a claim on the dividend stream of the corporation, in relative proportion to the outstanding shares. The aggregate value of a corporation's issued shares is its market capitalization.

Inputs into the financial market are from the firms issuing corporate bonds and shares to raise financial capital. The banking sector invests in stocks, corporate bonds and government bonds by holding asset portfolios that are invested in the financial market. The private sector (household sector and firms) also holds asset portfolios in stocks, corporate bonds and government bonds. In the model we abstract from the issuance of new securities (the primary equity market) and only consider the secondary equity market. We also abstract from financial market employment.

Government sector The government sector is composed of two parts: Government and a single Central Bank (the European Central Bank).

Government Government receipts mainly consist of taxes: income tax, corporate tax, sales tax and payroll tax. Furthermore, the government pays for public expenditures on education and infrastructure (both to the investment good producers) and public expenditures on R&D to the consumption good producers (government subsidies for training). The expenditures on education raise the general skill level of the working population, while the public expenditures on R&D are aimed at raising the technology-specific skill levels. The government pays out household transfers (unemployment and Social Security benefits). Finally, the Government issues government bonds either to the Central Bank or to the bond market directly, in order to finance the budget deficit. If there is insufficient demand for bonds on the bond market then the Central Bank will buy the remaining bonds from the Government, in exchange for fiat money. The Government pays out a bond coupon on the outstanding bonds. We abstract from government employment of administrators and other government officials.

Central Bank The Central Bank has as its main tasks to manage the monetary system and to maintain price stability. If a particular bank cannot meet its reserve requirements, the Central Bank issues reserves to the bank at a discount rate. If the Government issues government bonds while there is an insufficient demand for bonds on the bond market then the Central Bank will buy these bonds from the Government.

The devices for implementing a monetary policy by the Central Bank include contractionary, expansionary or stabilizing monetary policies:

- A contractionary policy means that if there is an excess demand for bonds on the bond market, the Central Bank sends the bond market a sell order. The Central Bank wants to sell bonds in order to extract money from the economy.

- A expansionary policy means that if there is an excess supply of bonds on the bond market, the Central Bank sends the bond market a buy order. The Central Bank wants to buy bonds in order to infuse money into the economy.

- A stabilizing monetary policy means that the Central Bank wishes to maintain a stable bond price by sending both buy and sell orders to the bond market.

Rest of the World (ROW) The ROW block consists of imports and exports to and from the EU block. We leave open the possibility to have multiple ROW blocks to model several important trade partners of the EEA: OPEC, US, Russia, India, Japan and China (see Fig. 5). The OPEC sells energy to the investment goods producers, to produce the investment goods. This implies that we model the EU as an open economy, taking into account the import and export flows of commodities and money.

4 Modelling guidelines

The aim of this section is to operationalize the abstract, low-level descriptions of the Modelling Requirements Document into working algorithms. We require algorithms for the four main blocks, i.e. the goods market, the labour market, the financial asset market and the credit market. A pseudocode example for a generic agent-based model may look something like shown below.

Pseudocode example for an economy loop

Algorithm 4.1 ECONOMY LOOP

```
function INITIALIZE
    Create trading environment (local, global, direct, indirect trading)
    Create market institutions (bilateral, centralized, semi-decentralized exchange)
    Create local interaction structures (networks, lattices, CA)
    Create agent population X, representing consumers/workers
    Create agent population Y, representing firms
    Create firms and allocate them randomly to sectors
    Create workers and allocate them randomly to sectors (human capital endowments)
end function

function LOOP ECONOMY
    for Period i = 1 : T do
        for each agent do                                  ▷ agent loop
            perform independent actions
            perform interactions
        end for
        compute statistics                                 ▷ collect statistics
    end for
end function
```

4.1 Goods market guidelines for use in WP5 - Proposed by Cagliari and Aix units

This section provides an algorithm for the goods market and the labour market, based on the description in Catalano et al. (2006).

A natural agent-based representation is to associate a software agent to each economic agent: worker-consumer, firm and bank. At each time step, the interactions among agents are of two basic kinds. The interactions consumer-firm to buy goods and firm-bank to get credit are similar (type 1 interaction), while the interactions worker-firm to find a job are different (type 2 interaction). To these interactions, we must add some computation made by each agent and requiring only its local data, and possibly access to global data. These computations are intrinsically parallel and asynchronous.

4.1.1 Interactions of type 1 (Goods and credit markets)

We assume two populations, X and Y, whose agents are x_i and y_j. In the case of the goods market, X consists of consumers and Y consists of firms. In the case of the credit market X consists of firms and Y consists of banks. The interaction is the following (see Fig. 2).

1. All agents of population X (x_i) randomly build a list of agents of population Y (y_j). This list includes only a subset of Y.

2. Every agent x_i queries all agents y_j in her list, retrieves a value from each agent y_j, and sorts her list according to these values. The sorting can be computed in parallel by all agents x_i.

   Remark 4.1 *This step could also be accomplished by first globally sorting all agents* $\{y_j\}$ *using the above quoted values, and then selecting the random list from this global information, preserving the order.*

3. Each agent x_i, taken in random order, queries the agents in her sorted list. Each agent y_j of the list is queried about some of its values, and immediately some of its values are updated. This change in state affects subsequent interactions of y_j.

   Remark 4.2 *Step 3 can be performed with a sequential polling of agents* $x_i \in X$*, provided that the order is random and changes at each time step. This solution leads to poor exploitation of the parallelism of the agent-based model.*

Alternatively, step 3 can be performed in parallel, provided that:

1. The first agent y_j of the sorted list of x_i is *locked*, then its relevant values are retrieved (in the case of the goods market, these are the offer price and the available quantity of the good), then some values of y_j are updated (in the example, the available quantity), and eventually y_j is *unlocked*. The computation proceeds for all agents in the sorted list, or until a condition on x_i is satisfied (in the example, when x_i has spent all its money).

2. The parallel interactions are in fact performed in such a way that they are equivalent to a random choice of agents $x_i \in X$, with no recurrent pattern.

4.1.2 Interactions of type 2 (labour market)

We again assume two populations, X and Y, whose agents are x_i and y_j. In the case of the labour market, X consists of workers and Y consists of firms. Each agent of type Y already holds a list of agents of type X that it is related to: the workers employed with the firm in the previous time step. The interaction is the following.

1. Every agent y_j decides, based on its status, how many agents x_i of its list to drop (to fire in the labour market example), or how many new agents to add to its list (to hire). In the former case, dropped agents are notified. In the latter, the agent y_j has to post vacancies.

2. Agents of population X which are not linked to any agent of population Y (unemployed workers in the example) randomly build a list of agents of population Y. This list includes only a subset of Y, and does not include agent y_j which just dropped the current agent x_i, if this is the case.

3. Non-linked agents x_i, in random order, queue the agents in their sorted lists. Each agent y_j in the list is queried as to whether it wishes to add any agent $x_j \in X$ to its list. If this is the case, x_i is added to the list, the number of vacancies of y_j is decreased by one, and x_i stops the computation. If not, x_i queries the next agent y'_j of its list, and so on.

4. Step 3 can be performed with a sequential polling of non-linked agents $x_i \in X$, provided that the order is random. Alternatively, it can be performed in parallel and asynchronously, provided that the first agent y_j of the sorted list of x_i is locked, then its relevant values are retrieved (the number of vacancies in its list), then possibly some values of y_j are updated (the number of vacancies and the members of its list), and eventually y_j is unlocked. The computation proceeds for all agents in the sorted list, or until x_i is linked to an agent $y_j \in Y$.

Type 2 interactions, except for the first simulation step, involve only a subset of agents of kind X. In our case, the status of the economy and the number of workers determine how small this subset is.

4.1.3 Possible extensions

In order to test the suitability of the X-agents approach to model a real economy, we might add geographical factors to the C@S model. This extension of the model could be:

1. Each firm has a geographical location, which is fixed.

2. Each consumer-worker has a geographical location, which is the same as the firm he is employed in.

3. Agents who work for the same firm plant must all occupy the same location.

4. When an employee changes employer, she must migrate/relocate to the new firm location.

4.1.4 Local interaction structure

The local search process on the goods and labour markets currently consists of consumers randomly sampling a fixed number of firms from the firm population. This sample of the population of firms then forms the consumer's 'local market' that it can use in its sequential shopping process. Such a ‘random-sampling-and-searching’ process can easily be extended to a model that has a more explicit interaction topology, for instance by using networks in combination with a geographical grid.

Consider firms and consumer-workers located on a 2-dimensional latice. The links between the workers and the firms could be modelled using an evolving network of labour market relations and consumer loyalty relations. The local search by a consumer for the produced output of firms now depends on its limited vision on the grid. Movements across the grid by both consumers and firms depends on their aspiration levels (the satisfaction/gratification level for the consumers and the aspired market sales level for the firms). Both the firms and the consumers are relocating when they cannot find ways to satisfy their aspiration levels. Using such a geographical grid could yield more insights into issues related to local interaction between the agents in the model.

Consumers who have been demand rationed during the previous period want to move to a neighborhood where there are more firms, in order to satisfy their demand. Firms who experienced unsold output at the end of the previous period may want to move to an area of the grid where there are more consumers to satisfy their desired sales targets. The consumers who have satisfied their satisficing demand, i.e., who have met their aspiration levels of consumption, will stay at their current location. Similarly, firms who did not experience any unsold stocks remain in the area where they were successful in selling their output.

The labour market is also a local interaction market with workers applying only for jobs in their local neighborhood. Firms post vacancies only locally and may be restricted to consider the workers’ proximity to the production facility, indicating that they are only considering local applicants for the posted jobs. The posted vacancies have some limited range across the grid. If workers cannot find a job, they move. If firms cannot find enough workers, they might move their production plants. But if firms cannot reach enough consumer demand, they might simply move their outlet stores to different regional markets.

The relocation of firms to a different area should not be taken literally in the sense that firms relocate their factory in every period. It simply refers to a firm’s outlet stores that can be relocated anywhere within a limited range of the firm’s current outlet stores (that is, firms have a limited range of vision as well). To model the relocation of outlet stores we introduce an new entity, the so called ‘outlet mall’ which is a local conglomeration of outlet stores of different firms in the region.

The relocation of the consumer-workers across the grid should similarly not be taken literally to mean that households relocate and move homes in every period. It just means that consumers can go shopping in different areas of the grid at different outlet malls that are further away, or that workers can apply to vacancies posted by firms in different parts of the grid, again taking into account that the consumer-workers have a limited range of vision.

4.1.5 Implementing the local interaction structure

Incorporating the local interactions described above into the search-and-trade procedures is relatively straightforward. On the goods market and the labour market, both firms and consumer-workers can do the search-and-trade process synchronously:

1. In the goods market, the list of firms a consumer considers for buying goods is influenced by proximity: closer firms have a higher probability to be included in the list. If a consumer cannot fulfill its ***satisfaction level*** it will look for firms elsewhere. This is a model of satisficing behavior.

2. In the labour market, the list of firms a worker considers to send its applications to is influenced by proximity: closer firms have a higher probability to be included in the list. If a worker cannot reach its ***job satisfaction level*** the agent may quit, or if its ***desired employment status*** cannot be reached and the agent remains unemployed, the worker can relocate and look for jobs elsewhere. This is a model of job satisfaction.
3. In the goods market, the list of consumers that a firm is considering for targeting its advertisement activities is influenced by proximity: closer consumers have a higher probability to be targeted. If a firm does not satisfy its ***sales targets***, it will intensify its advertisement activities and look for more consumers elsewhere. This is a model of advertising.
4. In the labour market, the list of workers a firm is considering to hire is influenced by proximity: closer workers have a higher probability of being included in the list. If a firm cannot find enough workers in close proximity to its plants, it will search for more workers elsewhere. This is a model of outsourcing.
5. Credit market interactions between the firms and banks, and other computations, do not depend on proximity. The credit market is assumed to be a centralized market, hence there is no decentralized interaction.

4.1.6 Transition probabilities

We might want to link the process of relocation of the agents in the labour market with the corresponding transition probabilities:

- When an agent is already employed by a firm:
  - there is a small probability of being fired, α (depending on the proportional contribution to overall team production).
  - there is a small probability of staying/being re-employed, β.
  - there is a small probability of quitting, γ (depending on job satisfaction).
- When an agent is unemployed:
  - there is a small probability of being hired, δ (depending on expected contribution versus wage demand).
  - there is a small probability of starting up a new singleton firm, ε.
  - there is a small probability of staying unemployed, $1 - \delta - \varepsilon$.
- When an agent starts up a new firm:
  - it remains in its present location, with high probability $1 - \epsilon$.
  - it selects a new location randomly, with small probability ϵ.
- When an existing firm goes bankrupt, what happens to the agents employed by the firm:
  - all agents become unemployed.
  - with small probability an agent remains in its present location, $p_{migrate}$.
  - with high probability an agent migrates to a new location selected at random, $1 - p_{migrate}$.

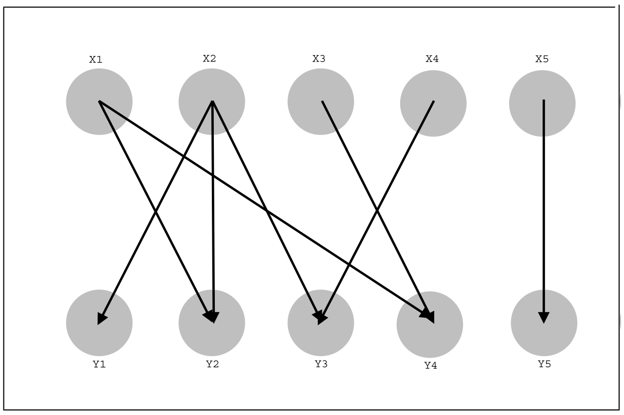

Figure 2: The querying protocol on the labour market. Workers (x_i) query the firms (y_j) for vacancies. Arrows indicate the direction of the querying messages being send. Workers can query multiple firms for vacancies at the same time, and a firm can have job applications from more than one worker pending consideration during the job market matching procedure.

4.1.7 Pseudocode examples for consumption goods and labour market models

Algorithm 4.2 LABOUR MARKET LOOP

function FIRM EMPLOYMENT LOOP
  for each agent $y_j \in Y$ **do** ▷ Employment by firms
    consult current employee list $List(y_j)$
    decide number of current employees to drop from list ▷ firing
    if number of employees to fire is positive **then**
      notify fired employees
      for agent to fire **do**
        $List(y_j) := List(y_j) - x_i$
      end for
    end if
    decide number of new employees to add to list ▷ hiring
    if number of new employees to hire is positive **then** ▷ post vacancies
      update y_j's internal value: increase number of vacancies by n_{vac}
    end if
    EmployNewWorkers
    if EmploymentLevel(y_i) = Unsatisfied **then** ▷ still vacancies to fill?
      open new plants
      move geographical location of plants ▷ move production
      post new vacancies for local employees ▷ search for more employees
    end if
  end for
end function

Algorithm 4.3 LABOUR MARKET LOOP

```
function WORKER EMPLOYMENT LOOP
    for each agent $x_i \in X$ do                                          ▷ Applications by workers
        if EmploymentStatus($x_i$) = unemployed then
            build random list of local firms $List(x_i) \subset Y$          ▷ list of local firms
            for each firm in the list $List(x_i)$ do
                if agent has just been fired by firm $y_j$ then
                    $List(x_i) = List(x_i) - y_j$                           ▷ do not re-apply with firm $y_j$
                end if
            end for                                                         ▷ definitive application list
            for each firm in the application list $List(x_i)$ do            ▷ apply with local firms
                lock $y_j$ for communication
                query firm $y_j$ for vacancies
                if firm $y_j$ has vacancy then
                    update $x_i$'s internal value: EmploymentStatus = employed by firm $y_j$
                    update $y_j$'s internal value: decrease number of vacancies by $1$
                    update $y_j$'s internal value: $NewList(y_j) := List(y_j) + x_i$
                    unlock $y_j$ for communication
                    set $List(x_i) = \emptyset$                              ▷ stop the application loop
                else
                    $List(x_i) := List(x_i) - y_j$                          ▷ continue with next firm
                    unlock $y_j$ for communication
                end if
            end for
            if EmploymentStatus($x_i$) = unemployed then                    ▷ still unemployed?
                move geographical location                                  ▷ relocate
                build new list $Y$ of local firms                           ▷ search for more firms
            end if
        end if
    end for
end function
```

Algorithm 4.4 GOODS MARKET LOOP

function CONSUMER TRADE LOOP
for each agent $x_i \in X$ **do** ▷ Seek firms for consuming activity
build a random list of local firms $List(x_i) \subset Y$ ▷ list of local firms
for each firm in the list $List(x_i)$ **do**
query firm y_j for internal values: offer prices, available units
end for
sort list $List(x_i)$ according to obtained internal values
while SatisfactionLevel(x_i)=Unsatisfied AND $List(x_i)$ non-empty **do**
for each firm in the shopping list $List(x_i)$ **do**
lock y_j for communication
query firm y_j for internal values: offer prices, available units
if firm y_j has units **then**
bargaining between x_i and y_j over price, quantity
buy product from firm y_j
update x_i's internal value: Inventory $= +1$ unit
update y_j's internal value: Inventory $= -1$ unit
end if
unlock y_j for communication
$List(x_i) := List(x_i) - y_j$ ▷ continue shopping with next firm
if SatisfactionLevel(x_i)=Satisfied **then**
set $List(x_i) = \emptyset$ ▷ stop the shopping loop
end if
end for
end while
if SatisfactionLevel(x_i)=Unsatisfied **then** ▷ still unsatisfied?
move geographical location ▷ search for more firms
build new list Y of local firms
end if
end for
end function

Algorithm 4.5 GOODS MARKET LOOP

function FIRM TRADE LOOP
    for each firm $y_i \in Y$ **do** ▷ Seek local consumers for advertising activity
        build a random list of local consumer groups $List(y_j) \subset X$
        while AspiredSalesLevel(y_j)=Unsatisfied AND $List(y_j)$ non-empty **do**
            for each consumer group in the list $List(y_j)$ **do**
                target advertisement to consumer group
                $List(y_j) := List(y_j) - x_i$ ▷ continue advertising with next consumer
                if AspiredSalesLevel(y_j)=Satisfied **then**
                    set $List(y_j) = \emptyset$ ▷ stop the advertising loop
                end if
            end for
        end while
        if AspiredSalesLevel(y_j)=Unsatisfied **then** ▷ sales targets still not met?
            move geographical location of current outlet stores ▷ search for more demand
            open new outlet stores at new malls
            build new list X of local consumers ▷ costumers at outlet mall
        end if
    end for
end function

4.2 Asset market guidelines for use in WP6 - Proposed by the Genoa unit

The most salient features for WP6 include listing the traders' characteristics and the types of assets to be considered in the asset market algorithm. For a more detailed model proposal, see Raberto et al. (2007).
The artificial financial market should include a number of heterogenous traders characterized by bounded rationality and different degree of information about the state of the economy, a number of risky assets, e.g., dividend paying stocks, a risk-free asset, and a particular market clearing mechanism. Traders are endowed with finite financial resources and are subject to an income stream. With respect to the tasks related to WP6, both the income streams and the dividend processes should be considered as exogenous and modelled according to well-defined stochastic processes, whereas with respect to the whole project they should be considered endogenous on the basis of models delivered by WP5 and WP7.
For example, the link between the financial asset market and the real production process of the firms should be addressed in close collaboration between Work packages 5, 6 and 7. For proposals in this direction, see Gallegati et al. (2007), and Raberto et al. (2007, Ch. 6)).
Also the link between the behavior of the consumers on the consumption goods market and the financial asset market should be addressed in these Workpackages.

4.2.1 Types of agents

The following types of agents will be active on the financial market.

Active Agents:

- Households: invests in assets (stocks, corporate bonds, government bonds).
- Firms: issues assets (stocks and bonds) and distributes its profits to shareholders.
- Asset Management Companies: a firm that manages Exchange Traded Funds (ETFs) and/or hedge funds. Like other firms, the AMC distributes its profits to its shareholders. There can be multiple AMCs.
- Government: issues government bonds.
- Financial Advisor: gives advice to households on the past performance of a set of portfolio allocation rules.

4.2.2 Traders' characteristics

Traders on the asset markets are characterized by the following features:

- their degree of knowledge about key economic variables.
- a set of internal variables describing the state of the trader.
- a set of decision variables.
- a set of beliefs about the (stochastic) processes underlying the financial dynamics.
- a structure of preferences.
- a set of behavioral rules.

- a set of parameters for tuning the behavior, using the learning algorithms.

Regarding the traders' degree of knowledge about key economic variables, this knowledge concerns:

- the risk-free rate of interest on government bonds.
- beliefs about the uncertainty structure of the economy.

The internal variables of a trader should include:

- Its financial portfolio and investment strategies.
- A historical record of price time series for the stocks the agent owns.
- The traders financial position: stock portfolio, bond holdings, savings account, current account.

Traders' decision variables should regard at least:

- allocation weights for every asset describing the investment portfolio choice.
- quantities and prices for each submitted limit-order or market-order.

Agents form beliefs concerning:

- assumptions concerning the dividend process.
- a subjective multivariate probability distribution of future price returns.
- or, alternatively, a subjective estimation of the expected average future returns and the expected variance-covariance matrix (if using a mean-variance framework).
- a subjective multivariate probability distribution of future dividends.
- or, alternatively, a subjective estimation of expected average future dividends (if using a mean-variance framework).

Estimates should be performed at different time horizons according to the learning algorithms specified in the EURACE Modelling Requirements (cf. Section 7 in van der Hoog and Deissenberg, 2007).

Concerning preferences and behavioral rules, we consider the following features:

- Decision variables should be set according to a preference rule, based either on expected utility theory on well-defined rules of thumb.
- A communication network among traders should also be considered in order to study how the information transmission process among agents may influence the price process and give rise to herding behavior and contagion effects on the financial markets.
- The modelling of traders' behavior should be based on well-known stylized facts about trader psychology.

4.2.3 Market clearing mechanism

The price formation process can be modelled either as a Clearinghouse mechanism or as a Limit-order Book (for detailed explanations of these mechanisms, see Section 6 in the EURACE Modelling Requirements, van der Hoog and Deissenberg, 2007). The clearinghouse mechanism requires all traders to submit their orders before a market price is determined, while the limit-order book mechanism allows for a continuous and asynchronous handling of orders. The latter is particularly suited for dealing with high frequency financial data. Most stock markets in the world nowadays employ limit-order books for managing transactions. However, since the EURACE project focuses on economic phenomena at time-scales greater than a business day, a clearinghouse mechanism seems more appropriate for the scope of the project. As an empirical motivation, it may also be noteworthy to note that this mechanism is used by the Taipei Stock Exchange in Taiwan at 20 second intervals to determine its asset prices.

Open issues

- With respect to the entire project, the asset market traders can be identified either with households trading directly on the market, or they can be identified with fund managers who manage the portfolios on behalf of the households.
- Concerning the types of assets to consider, see the remarks made in Section 3.3 on financial instruments. It is still an open question whether we should include derivative securities.

Proposed solutions

- The agents on the asset market are professional traders (brokers, dealers, investment banks). They are therefore market intermediaries for the households who invest in their stock portfolios, and the firms selling their stocks. One option is to have the households invest in a mutual fund that is managed by a fund manager who decides on the weights of the assets in the fund. The households can chose/change their fund manager infrequently, say monthly. But the fund managers re-shuffle the households asset portfolios, and are trading daily. The competition between different fund managers leads to different past performances, hence the incentive for households to switch between them. The way to deal with this is to model different portfolio allocation rules, and the households can switch between them. This is more fully described in Chapter 5 of Raberto et al. (2007) which is a proposal for introducing learning in the artificial financial market.
- Another issue is the introduction of the Asset Management Companies (AMCs) that trade Exchange Traded Funds, and the introduction of hedge funds. This is more fully described in Raberto et al. (2007) which gives the main outline of the financial market model.

4.2.4 Local interaction on the asset market

On many real-world asset markets floor-trading is being replaced with electronic trading because it is generally believed to be more efficient. Therefore it is not entirely clear what it means to introduce local interaction in the asset market model. So instead of local interaction between agents within a single asset market, we could think about having multiple asset markets that are locally distributed. Given that there is a geographical distribution of firms, we can introduce local asset markets on which only the stocks of local firms are traded. This implies that if a firm moves sufficiently far away, crossing a boundary, then their stocks will be traded on a different local asset market. But in general this outsourcing of firms will be rare, so most firms will stay in the same asset market.

Open issues

- If we want to implement such local asset markets, we have to decide on the resolution across the geographical grid. In other words, how many asset markets are there empirically in the EU-27? Alternatively, we could just consider local stock markets and leave the number arbitrary.

4.3 Modelling guidelines for use in WP7 - Proposed by the Bielefeld unit

In particular, WP7 is interested in studying potential interaction effects between innovative activity in firms and the supply characteristics of the labour force.

4.3.1 General features

In order to be able to appropriately address this research question which is spelled out in more detail in the Bielefeld unit's proposal (see Dawid et al., 2007), we aim at developing a model which looks at the interaction of a capital goods market, a consumption goods market, a labour market, as well as a credit and financial market, and an energy market. The energy market will be exogenous to the model simply serving the role as an input factor to the capital goods market. The interactions between the other markets will be more sophisticated. There will be a model of product and process innovation, as well as a labour market that hosts workers of various skill types. The link to the credit and financial market will ultimately be made via firms that may finance their investments externally, and via households that save part of their income. For proposals in this direction, see Gallegati et al. (2007) and Raberto et al. (2007).

4.3.2 Types of agents

The following three types of active agents and one type of passive agent (in the sense that this type of agent does not take any decisions) might be present in the model. Each type of active agent has several 'roles' corresponding to its activities in the different markets. The exposition focusses on the various roles played in the capital good, consumption good, and labour market, abstracting for the time being from activities in the credit and financial market. Each activity of an agent is connected to one of its roles. Regardless of its current role each agent can always access all its internal memory variables (like savings, available budget, stock of employees, skill level), therefore these internal memory variables represent the connection between the different roles of an agent.

Active Agents:

- Households
  - Consumption Goods Market: Role of Buyer
  - Labour Market: Role of Worker
- Investment Good Producers
  - Investment Goods Market: Role of Seller
  - Labour Market: Role of Employer
- Consumption Goods Producer
  - Investment Goods Market: Role of Buyer
  - Consumption Goods Market: Role of Seller
  - Labour Market: Role of Employer

Passive Agents:

- Outlet Malls
  - Consumption Goods Market: Information Transfer between Consumption Goods Producers and Households
- Market Research Entity:
  - Consumption Goods Market: Information Transfer to Consumption Goods Producers (market data about regional markets, i.e. market shares, local prices)
- Eurostat Entity
  - Labour Market: Sends information to firms about the wage level of highly skilled labour.

4.3.3 Decisions

In their various roles agents have to make decisions.

1. Capital goods producers:
   (a) Pricing the capital good
   (b) How much labour to hire
   (c) Production quantity
   (d) Investments in R&D
2. Consumption goods producers:
   (a) Production quantity
   (b) Technology choice
   (c) Investment in capital goods
   (d) How much labour to hire
   (e) At which mall to sell
   (f) How much to deliver at a mall
   (g) R&D investment for product innovation
   (h) Which price to set
3. Workers
   (a) Whether to search for a job
   (b) Accept or reject a job offer
   (c) Allocate budget on consumption and saving
   (d) At which mall to shop
   (e) Choice of consumption good

The various decisions listed have to be worked out in more detail. The aim is to apply where appropriate rules that are propagated in the 'management literature' as being applied in the real world. A proposal in this direction is made in Dawid et al. (2007).

4.3.4 Pseudocode example for a simple labour market model

The steps taken in the labour market are summarized in the following pseudocode. It serves as an example for the sequencing of functions that also has to be worked out in more detail for all other functions in the various markets to be modelled.

Algorithm 4.6 LABOUR MARKET LOOP

```
function LOOP LABOUR MARKET
    for n-times do
        Firms post vacancies
        Job seekers rank suitable vacancies and apply
        Firms rank applicants and send job offers
        Workers rank job offers and accept best offer
        Lists of vacancies and applications are adjusted for filled jobs
        if No successful match then
            Firms adjust wage offer
            Workers adjust reservation wage
        end if
    end for
end function
```

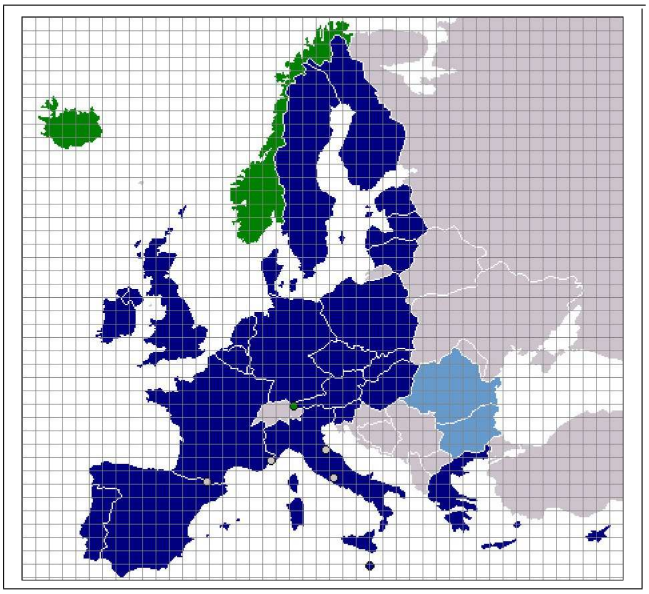

Figure 3: The EEA consisting of the EU-25, two accession countries and EFTA. The superimposed grid is approx. 100×100 km.

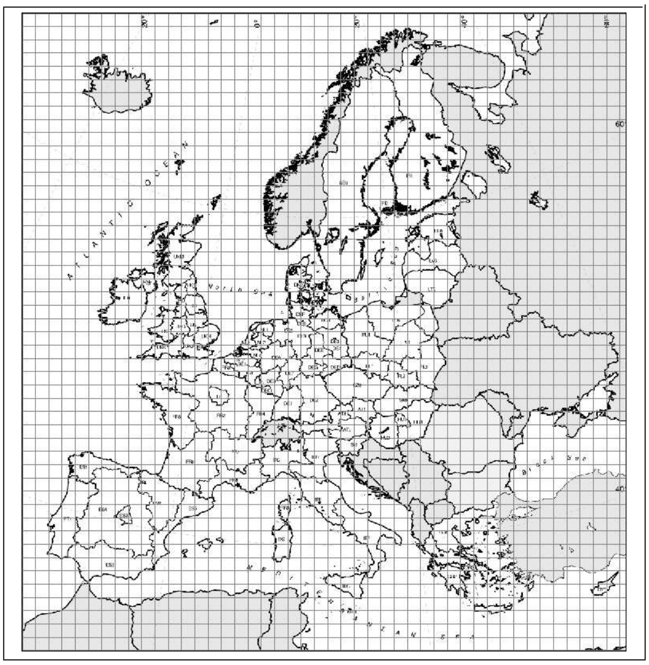

Figure 4: The EU-25 with NUTS-1 regions and the superimposed grid.

NUTS level classification.

Level	Minimum	Maximum
NUTS 3	$150,000$	$800,000$
NUTS 2	$800,000$	3 million
NUTS 1	3 million	7 million

Table 2: Population thresholds for defining NUTS levels.

NUTS regions.

	EU-25	Bulgaria	Roumania	Total
NUTS 0	25	1	1	27
NUTS 1	89	2	1	92
NUTS 2	254	6	8	268
NUTS 3	1214	28	42	1284
LAU 1	3334			
LAU 2	112,119			

Table 3: Number of NUTS regions in each NUTS level (data 2003).

EU-27 Main statistics.

population	$492,215,000$
area	$4,336,790\ km^2$
density	$115/km^2$
Grid size	$10,000\ km^2$
Required no. cells	434

Table 4: Geographical data on the EU-27 (IMF data estimates 2007).

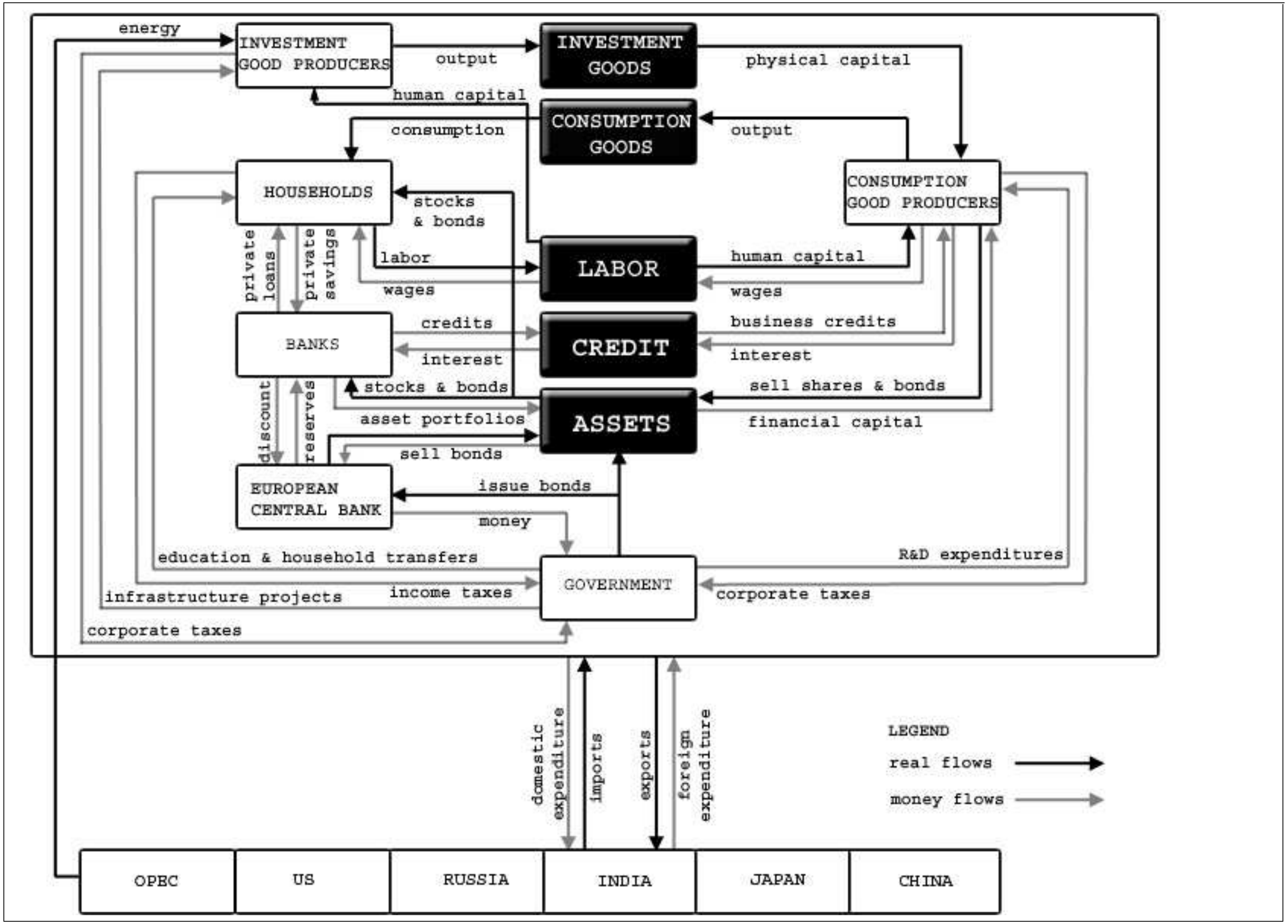

Figure 5: Diagram of the basic flows within an open economy.

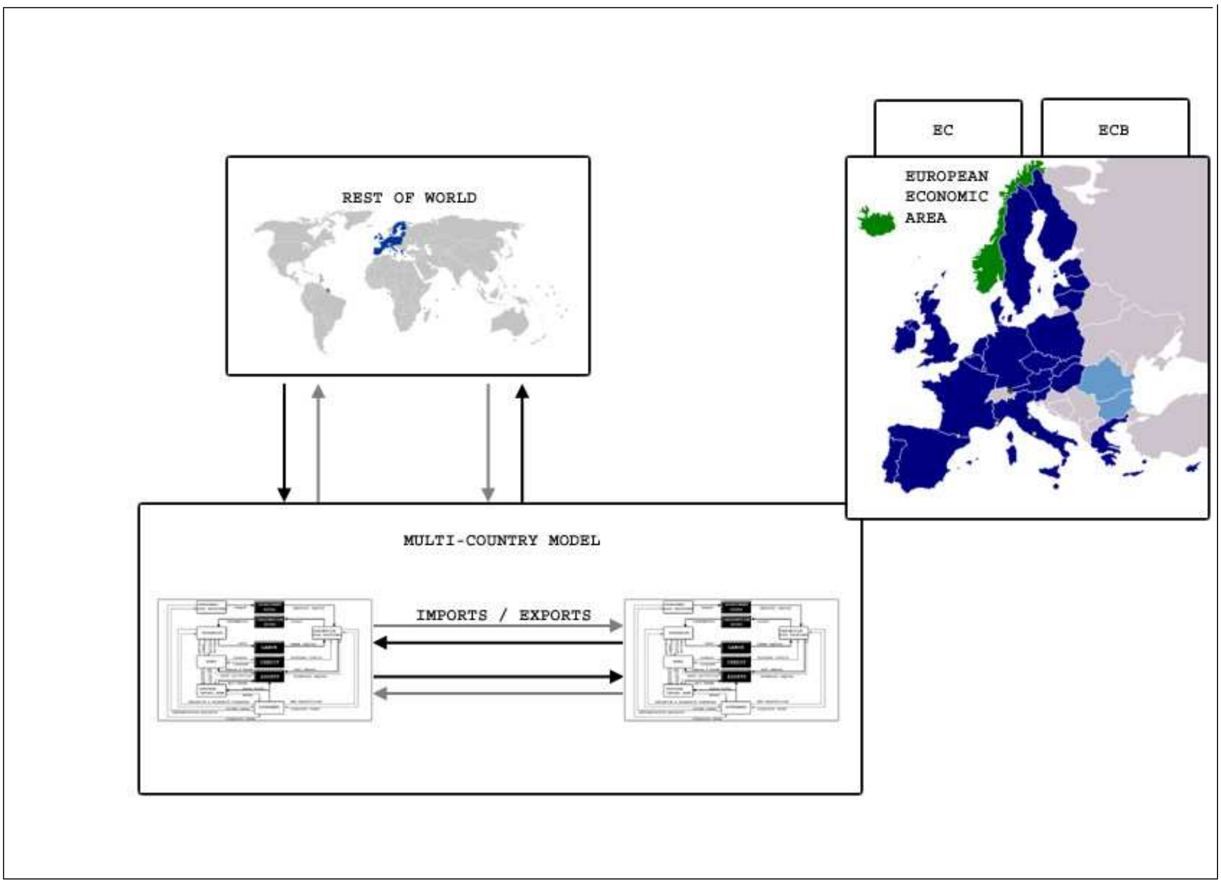

Figure 6: Diagram of the World Model.

References

Catalano, M., Clementi, F., Delli Gatti, D., Guilmi, C. D., Gaffeo, E., Gallegati, M., Giulioni, G., Napoletano, M., Palestrini, A., Russo, A., 2006. The C@S project. Version September 22, 2006.

Coakley, S., Kiran, M., 2007. X-Agent framework and software environment for agent-based models in economics. EURACE Report D1.1, Department of Computer Science, University of Sheffield.

Dawid, H., Gemkow, S., Harting, P., Kabus, K., Neugart, M., Wersching, K., 2007. Agent-based Models of Skill Dynamics and Innovation. EURACE Report D7.1, Bielefeld University.

Gallegati, M., Richiardi, M., Clementi, F., 2007. X-Agent-Based Models of Goods, Labour and Credit Markets. EURACE Report D5.2. Department of Economics, Università Politecnica delle Marche.

Holcombe, M., Coakley, S., Smallwood, R., October 2006. A general framework for agent-based modelling of complex systems. EURACE Working paper WP1.1, Department of Computer Science, University of Sheffield.

van der Hoog, S., Deissenberg, C., 2007. Modelling requirements for EURACE. EURACE Report D2.1, GREQAM, Université de la Méditerranée.

Raberto, M., Teglio, A., Cincotti, S., 2007. Agent-based models of financial markets. EURACE Report D6.1, DIBE, University of Genoa.

Tesfatsion, L., 2006. Agent-based computational economics: A constructive approach to economic theory. Handbook of Computational Economics, Volume 2: Agent-Based Computational Economics. Elsevier, Elsevier: North-Holland, handbooks in economics series 1, p. 55.