Modelling Specifications for EURACE

Responsible authors:
Sander van der Hoog (Aix-en-Provence)
Christophe Deissenberg (Aix-en-Provence)

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

Document acronym:	WP2-SH-2
Workpackage:	WP2.1 : Agent-based computational economics
Date:	October 1, 2007
Contributing units:	Bielefeld, Genoa, Cagliari, Sheffield
Responsible unit:	Aix
Comments requested from:	Outside experts
Document status:	4. Internal Report
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 4.2, Marco Raberto and Andrea Teglio contributed Section 4.3, and Herbert Dawid, Michael Neugart and collegues contributed Section 4.4. 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.

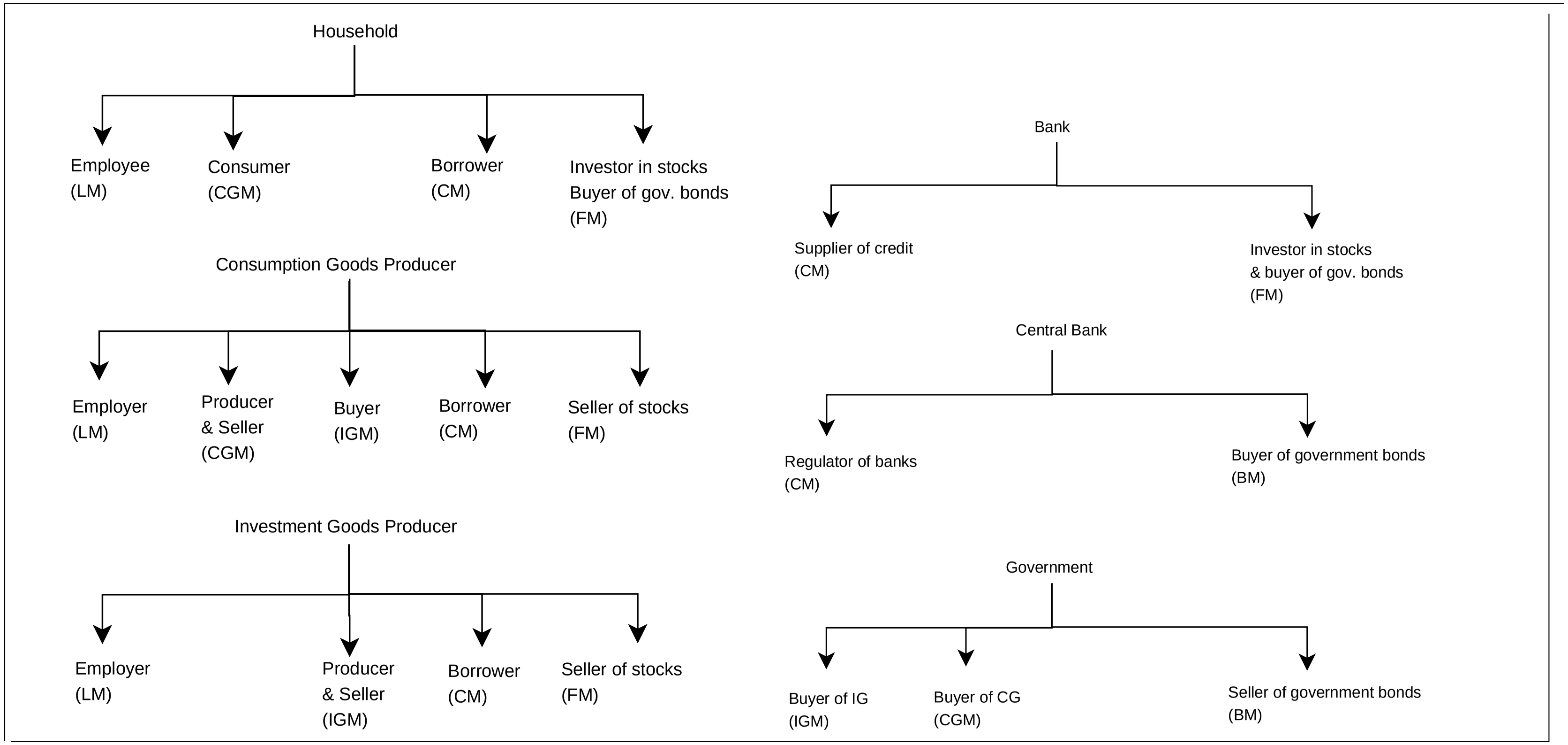

Figure 1: Diagrams of agent-role hierarchies highlighting the different market contexts in which the agents are active.

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, $1 - p_{migrate}$.

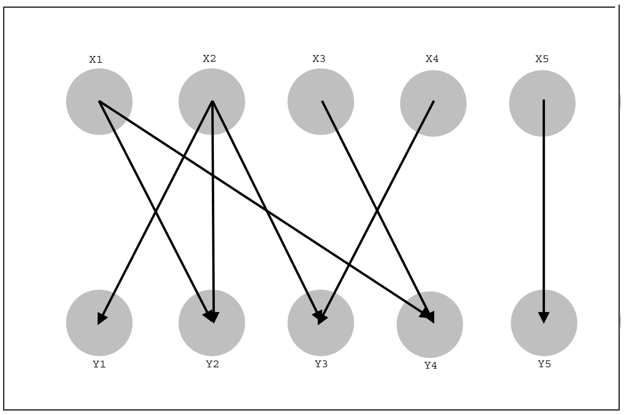

Figure 2: The querying protocol on the labour market. Workers (x_i) query the firms (y_j) for vacancies. Arrows indicate the direction of the querying messages being send. Workers can query multiple firms for vacancies at the same time, and a firm can have job applications from more than one worker pending 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 ∈ 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 ∈ X do                                          ▷ Applications by workers
        if EmploymentStatus(x_i) = unemployed then
            build random list of local firms List(x_i) ⊂ 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) = ∅                                  ▷ 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 ∈ X do                                    ▷ Seek firms for consuming activity
        build a random list of local firms List(x_i) ⊂ 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) = ∅                             ▷ 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 ∈ Y do                         ▷ Seek local consumers for advertising activity
        build a random list of local consumer groups List(y_j) ⊂ 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) = ∅                 ▷ 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 and Teglio (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 WP5 and WP6. For proposals in this direction, see Gallegati et al. (2007) and van der Hoog (2007b)).
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 van der Hoog (2007a) 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 and Teglio (2007) which gives the main outlines of the asset 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), Raberto and Teglio (2007), and van der Hoog (2007b).

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
```

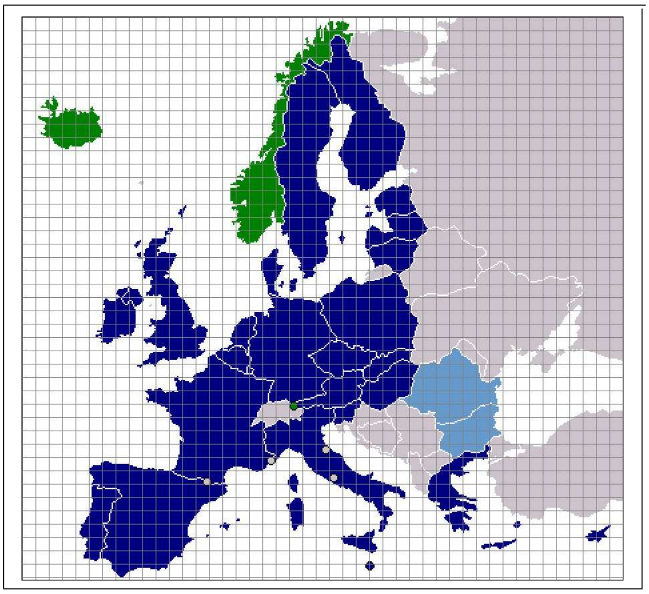

Figure 3: The EEA consisting of the EU-25, two accession countries and EFTA. The superimposed grid is approx. 100×100 km.

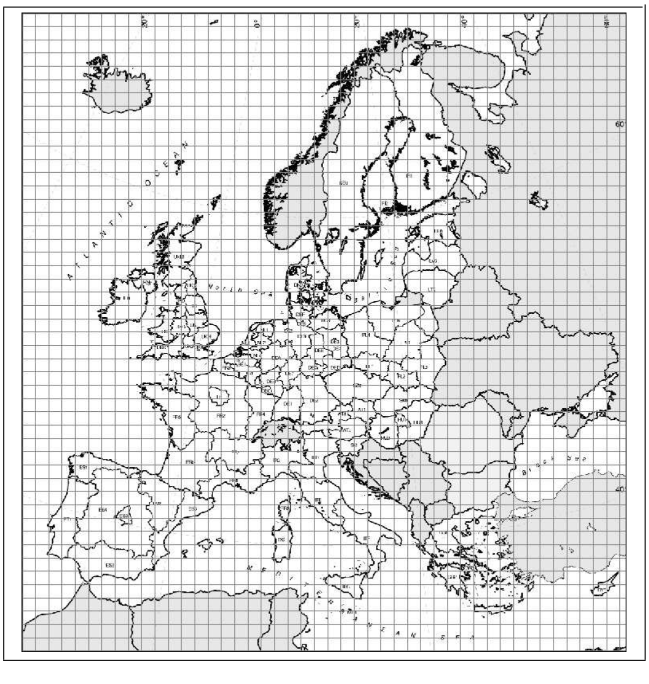

Figure 4: The EU-25 with NUTS-1 regions and the superimposed grid.

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).

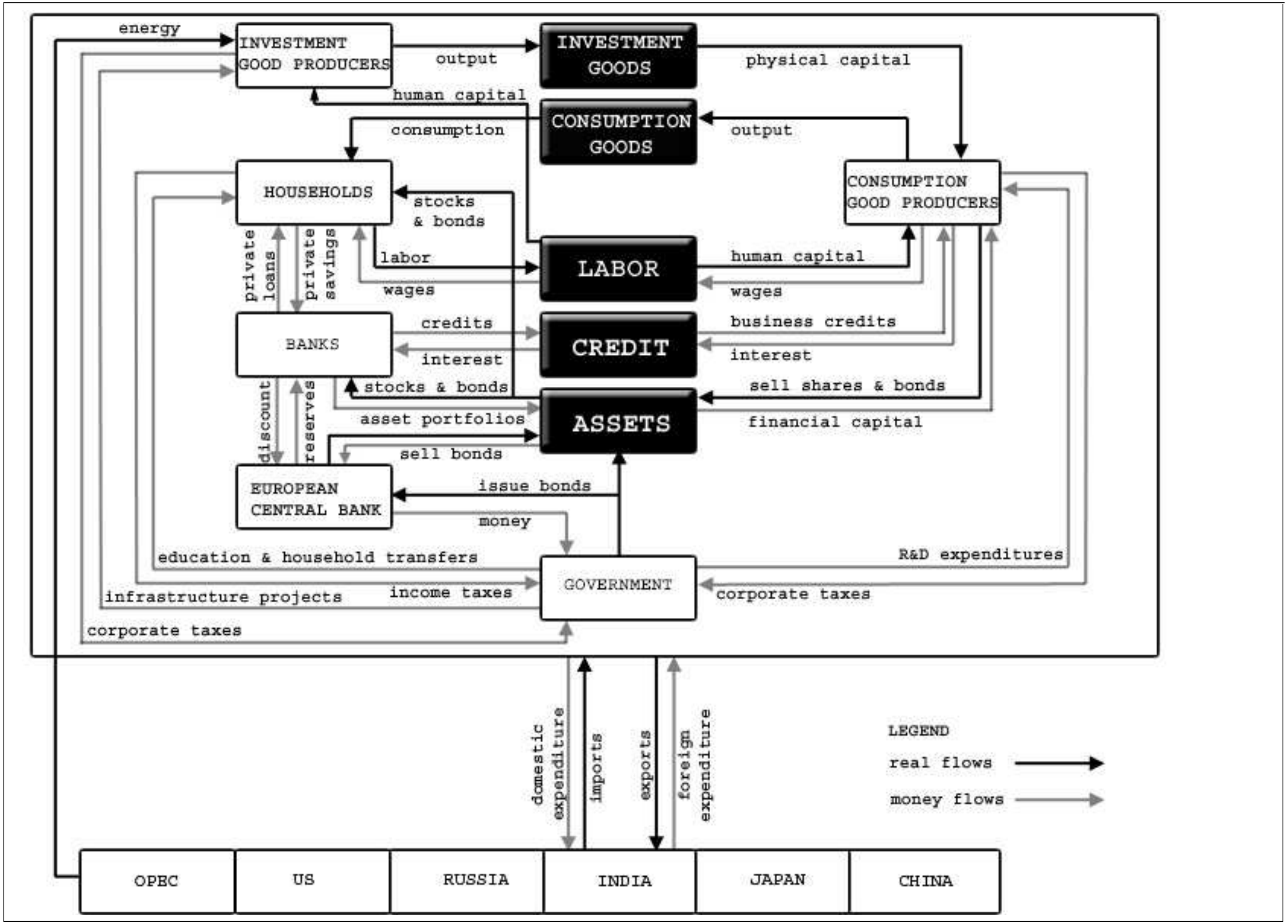

Figure 5: Diagram of the basic flows within an open economy.

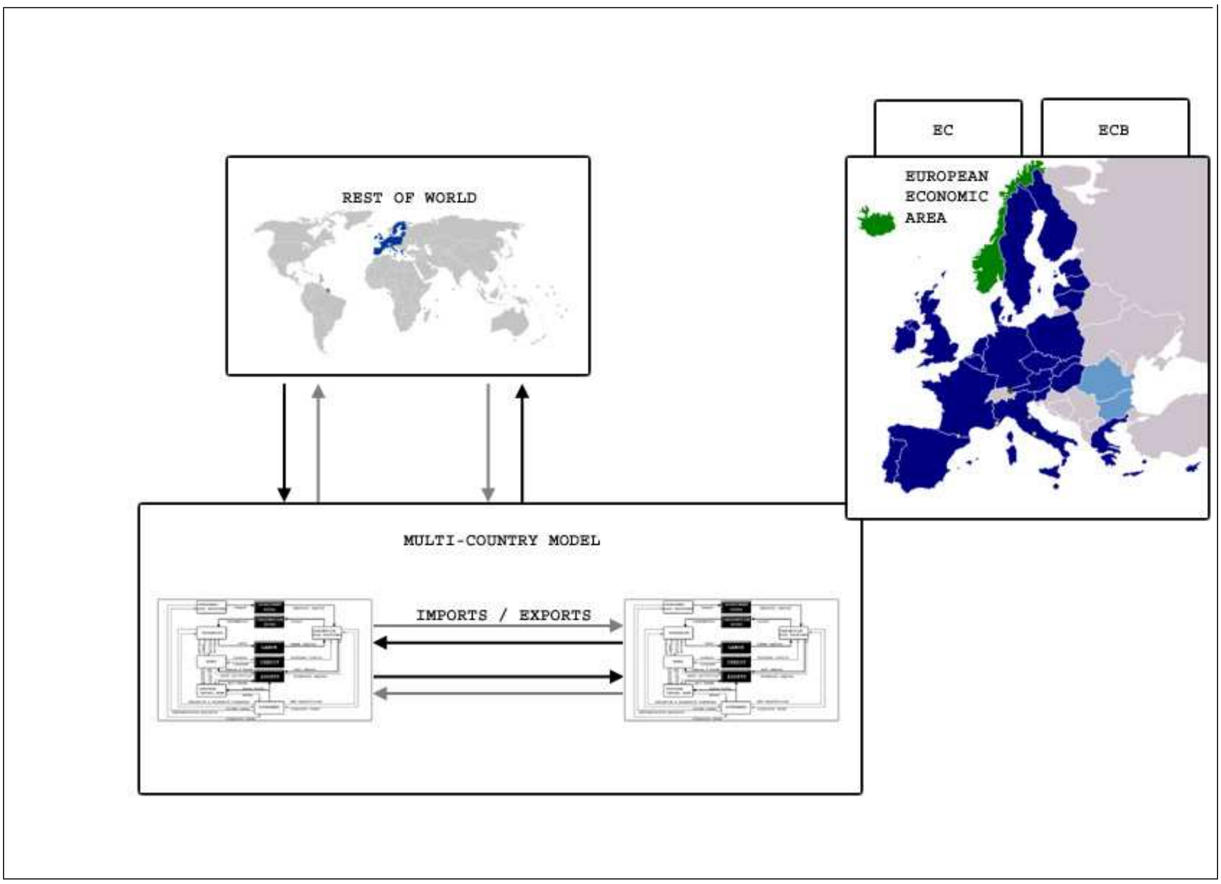

Figure 6: Diagram of the World Model.

References

Catalano, M., Clementi, F., Gatti, D. 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. EURACE Report D1.1: X-agent framework and software environment for agent-based models in economics. version September 2007, Department of Computer Science, University of Sheffield. Available online: http://www.eurace.org/Wiki/index.php/Categories/WP1:_Agent-based_software_engineering.

Dawid, H., Neugart, M., Wersching, K., Kabus, K., Harting, P., Gemkow, S., 2007. Capital, Consumption Goods and Labor Markets in EURACE. EURACE Working paper WP5.1, Universitaet Bielefeld, version April 20, 2007. Available online: http://www.eurace.org/Wiki/images/7/70/Goods_labor_markets_draft_unibi_0407.pdf.

Gallegati, M., Richiardi, M., Clementi, F., 2007. Modelling the Credit Market in EURACE. EURACE Working paper WP5.2, version July 16, 2007. Department of Economics, Università Politecnica delle Marche. Available online: http://www.eurace.org/Wiki/images/c/c9/CreditMarket_7.0.pdf.

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., 2007a. A Note on Modelling Learning in the M-AFM/X-AFM. EURACE Working paper WP6.3, version July 15, 2007. Available online: http://www.eurace.org/Wiki/images/a/af/LearningCS-v.3.pdf.

van der Hoog, S., 2007b. Financial Policy Decisions in EURACE. EURACE Working paper WP6.2, version August 1st, 2007. Available online: http://www.eurace.org/Wiki/images/0/0b/FirmFinancialPolicy-v2.pdf.

van der Hoog, S., Deissenberg, C., 2007. Modelling requirements for EURACE. EURACE Working paper WP2.1, version July 27, 2007. Available online: http://www.eurace.org/Wiki/images/9/98/WP2_Modelling_Requirements_Rev24072007.pdf.

Raberto, M., Teglio, A., 2007. The financial side of EURACE. EURACE Working paper WP6.1, version May 25, 2007. Available online: http://www.eurace.org/Wiki/images/2/25/Finance_wiki.pdf.

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.