

## EURACE: A massively parallel agent-based model of the European economy

Christophe Deissenberg<sup>a,\*</sup>, Sander van der Hoog<sup>a</sup>, Herbert Dawid<sup>b</sup>

<sup>a</sup> GREQAM, Université de la Méditerranée, Château Lafarge, Route des Milles, 13290 Les Milles, France

<sup>b</sup> Department of Business Administration and Economics, Bielefeld University, Universitätsstrasse 25, D-33615 Bielefeld, Germany

### ARTICLE INFO

#### Keywords:

Agent-based computational economics

X-machines

Parallel computation

### ABSTRACT

EURACE is a major European attempt to construct an agent-based model of the European economy with a very large population of autonomous, purposive agents interacting in a complicated economic environment. To create it, major advances are needed, in particular in terms of economic modeling and software engineering. In this paper, we describe the general structure of the economic model developed for EURACE and present the Flexible Large-scale Agent Modelling Environment (FLAME) that will be used to describe the agents and run the model on massively parallel supercomputers. Illustrative simulations with a simplified model based on EURACE's labor market module are presented.

© 2008 Elsevier Inc. All rights reserved.

### 1. Introduction

Research in economics has traditionally been and, to a large degree, still is based on the development and analysis of highly stylized, analytically tractable models. However, thanks to recent developments in computer technology and numerical methods, large-scale simulations are increasingly providing a powerful and attractive new approach for understanding the characteristics of economic systems and to derive economic policy recommendations. In particular, by explicitly modeling the decentralized interaction of heterogeneous economic agents in systems like markets or organizations, *agent-based computational economics (ACE)* attempts to transcend the numerous restrictive assumptions underlying most main-stream analytical models (e.g. homogeneity of individuals, perfect rationality, rational expectations, perfect ex ante coordination in an equilibrium).

The idea is simple enough. To build a ACE model, one creates an artificial landscape, possibly capturing real geographical features. This landscape is covered with factories, shops, schools, transportation and communication networks, natural resources, etc. It is populated with purposive agents that move around the landscape, communicate with other agents, work, consume, learn, invest, speculate on financial markets, that is, that potentially conduct all human activities of interest. The modeler specifies the landscape, the rules governing the landscape dynamics, the interaction among agents, between agents and the environment, the agents' behavior, and sets the initial conditions. He then lets the model evolve on its own, keeping track of the system output (i.e. chiefly of the individual actions and states) at any desired level of detail. Thus, the collected data can be used for the usual explanation/prediction/policy-making purposes.

Models of this kind have been developed in many areas of economics. Among others, they have been used to study (i) the emergence of trading behavior on goods-markets [1,2] and on financial markets [3,4], (ii) bidding behavior in auctions [5,6], (iii) numerous issues concerning innovation and industry evolution [7], or (iv) the emergence of cooperative behavior in

\* Corresponding author.

E-mail addresses: [christophe.deissenberg@univmed.fr](mailto:christophe.deissenberg@univmed.fr) (C. Deissenberg), [svdhoog@gmail.com](mailto:svdhoog@gmail.com) (S. van der Hoog), [hdawid@wiwi.uni-bielefeld.de](mailto:hdawid@wiwi.uni-bielefeld.de) (H. Dawid).

economic systems [8,9]. For an overview of main developments in ACE research over the last 15 years see Ref. [10]. The importance of the field is documented by numerous recent special issues in high-level journals [11–14] as well as the fact that an entire volume of the highly influential North-Holland Handbook series is dedicated to agent-based computational economics [15].

Most of the existing models, however, cover only a single industry, one restricted geographical area, or a unique market, and involve relatively small populations of agents. In contrast, the research project EURACE we are presenting in this paper aims at creating an agent-based model of the whole European Union, to be populated with a vast amount of fairly sophisticated agents on a complex landscape. By its scope and complexity, the effort is unsurpassed and needs to cover much *terra incognita*, among others concerning the conceptual and computational architecture of the model, its numerical implementation, its validation, and the exploitation of the simulation results. In particular, running such a large model will necessitate using massively parallel computing on large supercomputers, using pioneering software.

The 3-year EURACE Project started in September 2006. It includes economists and computer scientists from eight research centers in Italy, France, Germany, the United Kingdom, and Turkey, as well as the 2001 Nobel laureate in economics, Joseph Stiglitz (Columbia University). More institutional and scientific details can be found on the project's web page: [www.eurace.org](http://www.eurace.org).

This paper documents the state of EURACE with regard to the general structure of the economic model and the basic computational implementation. It illustrates the current work with an exploratory numerical investigation conducted with a grossly simplified model based on EURACE's labor market module. Important aspects such as the calibration, validation, and exploitation of the model are not discussed.

The paper is organized as follows. In Section 2 we give an overview of the economic model that will be used within EURACE. Section 3 describes the computational framework FLAME in which the model will be implemented. Section 4 provides some simulation results for the labor market, and Section 5 concludes.

## 2. The economic model

The model includes various artificial markets for real commodities (mainly, consumption goods, investment goods and labor) and markets for financial assets (such as debt securities, bonds and stocks). The artificial markets are first developed and studied separately, and integrated into a unified framework at a later stage.

### 2.1. Time and space

The final model aims at representing, albeit in a very simplified and stylized way, the EU-27. To that effect, we intend to link GIS data to economic data available from Eurostat at the so-called NUTS-2 regions level. NUTS are the geographical subdivision of the EU that Eurostat (the statistical office of the EU) uses when collecting its statistics. There are 268 NUTS-2 regions in the EU-27. This will allow us to distribute the firms, households, shopping centers, and other relevant economic elements over the EU-27 territory in a way that roughly reproduces reality. The rest-of-the-world is modeled as a separate, highly aggregated entity. It provides important (semi)exogenous inputs to the EU-27 model, such as energy prices.

The temporal resolution of the model is the business day. All economic activities can, but does not necessarily, take place on a daily basis. The various markets function at different time-scales, with buying and selling on the stock market taking place at shorter time intervals than, for example, the interactions on the job market.

Most actions are event-driven: an agent will take this or that action as a function of its current state and of the past history of all inputs received from its physical and socio-economic environment. However, we also allow for calendar-driven activities.

In addition to the geographical structure, the agents are linked by a number of socio-economic networks reflecting, for example, buyers-sellers, firms-workers, or banks-firms links. With agents interacting in parallel on different markets multiple layers of networks will be operational simultaneously. The networks are evolving over time, driven by the (dis)satisfaction of the agents with their current links, their exploration of alternative links, the emergence of new needs, and similar factors.

### 2.2. Agents and markets

There are three types of agents with learning capabilities: households (up to  $\sim 10^7$ ), firms (up to  $\sim 10^5$  producing consumption goods, and up to  $\sim 10^2$  producing investment goods), and banks ( $\sim 10^2$ ). Other agents, that is, national governments and the single central bank, follow simple, predefined decision rules, thus allowing a comparative analysis of the economic behavior under alternative policy regimes. Finally, there are a number of institutional agents whose main function is to gather and distribute information, to compute aggregate indices and economic indicators and to transmit these to selected groups of agents if these agents request the information. For example, we envision incorporating a statistical office mimicking Eurostat, and a market research entity that investigates the profitability of firms in local regional markets and transmits this data to firms who want to enter into a local market.

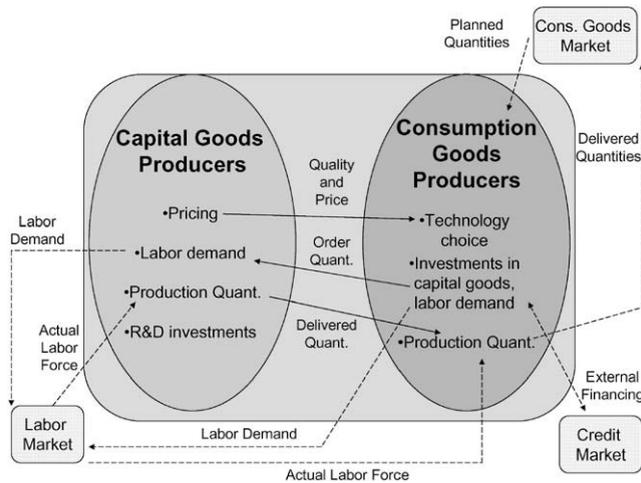


Fig. 1. Interactions between the investment goods producers and the consumption goods producers on the market for capital goods. Arrows show the messages between agents.

The model considers five types of markets: consumption goods, investment goods, labor, credit, and financial assets. With the exception of the investment goods market and the asset market, the markets are local. There is, for example, a local labor market in each NUTS-2 region. The local markets are interrelated, but typically through indirect or weak links (in the case of the labor markets, the main link is labor mobility).

The market for consumption goods is a decentralized market, with local interaction between the firms and consumers. We assume that the firms send their merchandise to a given set of local shopping malls. All buying and selling occurs at these malls. Firms chose the outlet malls on the basis of expected local demand and profit opportunities. They also take into account the costs involved in servicing a particular mall, such as the transportation costs, the leases for the stores in the mall, and the inventory management costs.

The labor market is also a decentralized market. A local search-and-matching process is used to represent the interaction between firms and workers. The firms post vacancies, including the minimum skill level required for the posted job. The potential employees apply to vacancies that have been posted by firms in their local neighborhood. Unemployed workers who do not succeed in finding a job locally can migrate to a different region.

The market for investment goods is a centralized market. There are multiple investment goods producers, each producing a different, vertically differentiated, technology. The investment goods producers invest in R&D to technologically improve the investment goods, leading to oligopolistic competition among them. The producers of consumption goods can invest in one of these technologies to produce a variety of differentiated consumption goods.

On the credit market, the firms interact with banks to obtain loans. The credit market is a decentralized market, with competition between banks setting different interest rates for the business loans. The banks apply credit standards to the firms that apply for the loans. Thus, the firms can be credit constrained.

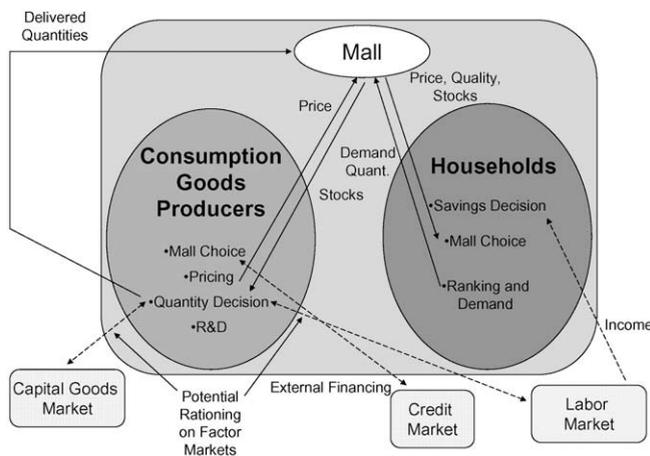


Fig. 2. Interactions between the consumption goods producers and the consumers on the goods market.

Finally, the financial asset market links the real side with the financial side. Firms issue equity (common stocks and corporate bonds) to finance investments and production. The households invest in asset portfolios, and the government sells government bonds to finance its budget deficit. The financial market thus consists of a market for corporate and government bonds and a market for firm stocks. The linkage between the financial side and the real side of the economy is provided by the financial policy of the firms on internal and external financing, that is among others, by the dividend, the debt repayment, and the investment decisions [16].

Figs. 1 and 2 show the interactions on the markets for investment goods and consumption goods between the producers of investment goods (new technologies), producers of consumption goods, and consumers.

### 3. Computational framework

Economic agent-based systems are intrinsically massively parallel computational systems with very large populations of sophisticated agents – meaning that up to millions of agents perform complicated local computations and exchange very large amounts of data with other agents. Thus, it appears natural and is indeed necessary to implement them on parallel supercomputers. Very few of the existing computational approaches, however, are truly adapted to the needs of such models. In particular, they rarely take into account the fact that the agents have precise positions on physical and conceptual networks – they are located in space and also linked in a precise manner through trade, contracts, friendship, and other socio-economic relationships. An exception to this is the FLAME framework, which will now be briefly presented.

#### 3.1. FLAME

EURACE will be implemented using the Flexible Large-scale Agent Modelling Environment (FLAME) developed by Simon Coakley, Mike Holcombe, and others at the University of Sheffield (see <http://www.flame.ac.uk> for a more complete presentation and references). FLAME's origins lie in an agent-based project on the simulation of biological cells grown under different environmental conditions. A key aim of the project was to write specifications for a formal framework allowing modelers to easily create, exchange, include and couple models that have been written in a high-level modelling language. Other key aims were the development of parallelization techniques, the distribution of agents over many processors, and the inclusion of testing methods to verify developed models. All these elements are vital to agent-based models in general and to EURACE in particular.

FLAME is based on so-called finite-state machines, that is, on automata described by a finite number of states, transitions between those states, and actions, that are heavily used in computational sciences [17]. The approach taken in FLAME is to regard each individual agent as a X-Machine and to specify a communication structure such that the different agents can exchange messages with each other. In other words, individual X-Machines are given the ability to communicate by exchanging messages. Moreover, they are generalized by providing them with an internal memory, leading to a so-called stream X-Machine design. The framework has been adapted to enable it to run on a parallel computing platform by Coakley [18]. It has been previously used to study the behavior of a number of biological systems – at the molecular, cellular, tissue and social levels – and has been successful in uncovering a number of new biological properties that have been confirmed experimentally by Coakley et al. [19].

In the following section, we present the computational model of stream X-Machines in more detail.

#### 3.2. Stream X-Machines

Stream X-Machines [20] are a variant of the X-Machine concept that was introduced by Eilenberg [21]. Like a finite state machine, an X-Machine consists of a finite set of states  $S$ , a set of transition functions  $F$ , and a language  $X$  used by the system to read and write information. However, an X-Machine also has an internal memory that influences the operation of the machine. The internal functions take as inputs internal memory variables and messages that are sent by other X-Machines. The output of an internal function is a modified value of an internal memory variable and/or output messages to other X-Machines.

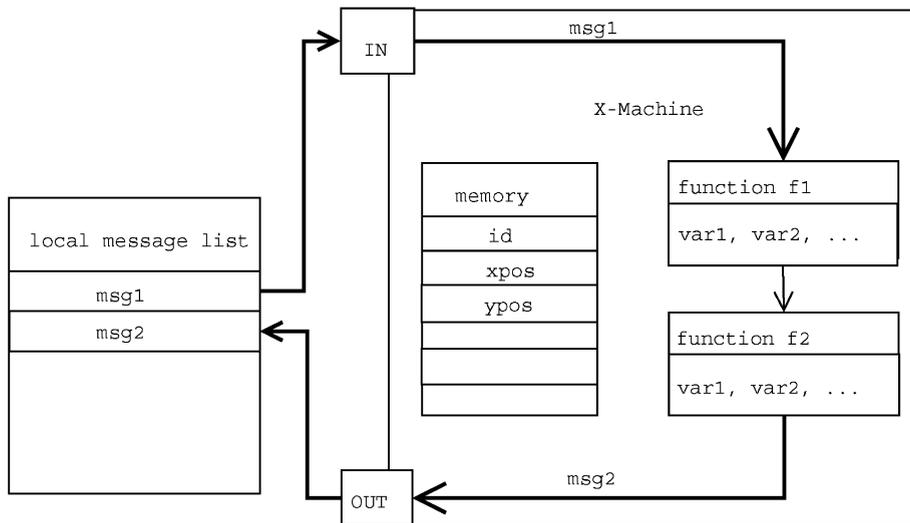
Stream X-Machines form a very general computational framework. For example, it is a simple matter to interpret Turing machines, the standard model for digital computation, as stream X-Machines. They allow us to treat in very general terms both the language and the computing of functions over that language, and thus make it possible to go well beyond the traditional limits of computational theory. In particular, the framework has proved itself most appropriate for describing and running large-scale agent-based computational models. Formally, a stream X-Machine is defined as follows.

**Definition 3.1.** A stream X-Machine is a 6-tuple

$$X = (M, m_0, \Sigma, \Gamma, F, \omega) \quad (1)$$

consisting of the following:

- a finite set of memory states  $M$ ,
- a start state  $m_0$  (also called the initial state) which is an element of  $M$ ,
- a finite set of symbols called the input alphabet ( $\Sigma$ ),



**Fig. 3.** Example definition of an X-Machine. A simple machine is defined, with internal memory variables  $id$  and position variables  $(x, y)$ . The machine has two functions,  $f1$  and  $f2$  that have an internal function dependency: function  $f1$  needs to run before  $f2$ . An input message  $msg1$  is needed as input to  $f1$ . The X-Machine may send a message  $msg2$  to communicate its current position  $(x, y)$ .

- a finite set of symbols called the output alphabet ( $\Gamma$ ),
- a state-transition function  $F: M \times \Sigma \rightarrow M$ ,
- an output function  $\omega: M \times \Sigma \rightarrow \Gamma$ .

The state of a stream X-Machine is entirely determined by its internal memory. Hence, the finite set of states  $S$  coincides with the set of internal memory states  $M$  of the X-Machine. The current memory state  $m \in M$  describes the machine's current information set. The language  $X$  is specified by the alphabets  $\Sigma$  and  $\Gamma$  that are symbol sets to encode input and output strings (messages). The state-transition function  $F$  of a stream X-Machine takes as inputs the current internal memory state and a (list of) input message(s); this determines the next memory state. The output function  $\omega$  is also a function of the current memory state and input messages, but has as its co-domain a (list of) output message(s) to other X-Machines.

Fig. 3 shows an example definition of an X-Machine, defining internal memory variables  $id$  and position variables  $(x, y)$ . The machine has two functions,  $f1$  and  $f2$ , and it can send out a message  $msg2$  containing its current position.

Benchmark tests of the FLAME framework have been performed on a number of parallel computers, including: SCARF<sup>1</sup>, HAPU<sup>2</sup>, NW-GRID<sup>3</sup>, and HPCx<sup>4</sup>. The test model consisted of  $10^6$  simple agents who only communicate their  $(x, y)$  position. The results are presented in Fig. 4, which shows how the 'time per iteration' decreases when the number of processors is increased.

In the following, the terms '(stream) X-Machine' and 'agent' are used equivalently.

### 3.3. Messages

As previously mentioned, FLAME regards each individual agent as an X-Machine and specifies a communication structure such that the agents can exchange messages with each other. All interdependencies between the agents' activities run through the messages, that is, there is no direct link between the internal functions of two separate agents. In this way, the content of the internal memory of each agent is shielded from outside access, so that all information is in a sense private information. An agent's behavior only depends on its internal memory state. The internal memory can for example include the agent's location or any other internal information that might change over time and affect the agent's behavior.

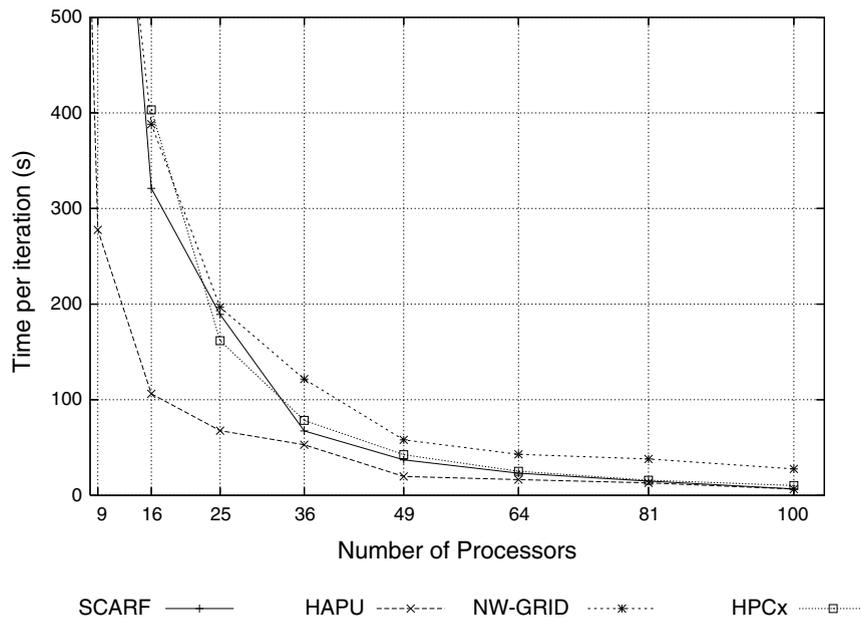
Exchanging messages, however, can be very costly on a parallel supercomputer. Schematically, such a machine consists of individual computers (the so-called computing nodes), that are linked by a high-speed communication network. The speed of communication among nodes is much smaller than the speed of communication within a node. Furthermore, a node may have to wait for a message from another node before being able to continue with its computations. Thus, running a large-scale agent-based computational model efficiently implies favoring intra-node computation over inter-node message exchanges.

<sup>1</sup> <http://hpcsg.esc.rl.ac.uk/scarf/index.html>

<sup>2</sup> <http://request.dl.ac.uk/hosts/hapu.live>

<sup>3</sup> [http://request.dl.ac.uk/hosts/NW\\_GRID.live](http://request.dl.ac.uk/hosts/NW_GRID.live)

<sup>4</sup> <http://www.hpcx.ac.uk>



**Fig. 4.** Performance of the FLAME framework on four different supercomputers, using varying numbers of processors. The plot shows how the 'time per iteration' decreases when the number of processors is increased. The test model consisted of  $10^6$  simple agents who only communicate their  $(x, y)$  position to each other.

To achieve that goal, we exploit the fact that in an agent-based model most information provided by one agent is sent to a relatively small group of neighboring agents (where 'neighbor' is meant in the functional rather than a geographical sense: two agents can be neighbors in a social network even if they reside on different continents). Thus, we split the landscape into different regions consisting of clusters of neighboring agents. The computations for each region are dealt with by a dedicated node, that is, they are carried out most efficiently. Migration of agents across the boundary of a region is dealt with by moving the agent's memory from one node to another.

The multi-agent model on a parallel machine can then be visualized as consisting of a population of agents inhabiting different neighborhoods (i.e. nodes) and who occasionally change neighborhood. Each neighborhood has its own message list that the agents can use to communicate making also communication within the neighborhood most efficient. When an agent sends a message it does not send the message directly to another agent, but sends it to the local message list.

A message received on a list is either read or ignored by the individual agent, as a function of the agent's internal rules. These rules can reflect the fact that, in reality, certain agents may not have access to certain types of messages. Private communication between two agents is made possible by including in a given message the *id* of the recipient and specifying internal rules that forbid any other agent from reading the message.

If an agent associated to a message list needs to send a message to an agent associated to a different message list, then the message is sent from one message list to the other. Such communication across message lists can be computationally costly, but the cost can be minimized by a proper distribution of the agent population across the different nodes in the computational cluster.<sup>5</sup>

Sending messages only to the list of one's own neighborhood is too restrictive in many cases. If an agent is close to the edge of its neighborhood and may affect an agent in an adjacent neighborhood, a copy of the message is also sent to the list of the adjacent neighborhood.

### 3.4. The definition of economic X-agents

In an economic agent-based implementation, the following items have to be made precise for every agent within the above general structure:

- the markets on which the agent can be active;
- the behavioral rules followed by the agent (these are captured by the internal functions  $f$  of the X-Machine);

<sup>5</sup> The protocol that is used to send messages is MPI. The system of local message boards can be likened to an e-mail system (i.e. the simple mail transfer protocol, SMTP), which operates by first sending all local e-mails to the local mail server. If the recipient's domain name, that is, that part of the e-mail address to the right of the @-sign, is the same as that of the sender, the message is handled internally. If not, the e-mail is sent to a router, which then relays the message to a different local mail server on the recipients domain.

- the allowable actions that the agent can perform and the decisions that the agent needs to make on each market (these are the changes to the internal memory variables, as a result of the application of the internal functions  $f$ );
- the messages that the agent can receive from other agents (these are the input messages  $\sigma \in \Sigma$  of the X-Machine);
- the messages that the agent can send to the other agents (these are the output messages  $\gamma \in \Gamma$  of the X-Machine).

### 3.4.1. Roles and contexts

We shall view a market as providing a *context* for agents to act in. Agents always act within contexts, but can have different *roles* in the same or different contexts. Accordingly,

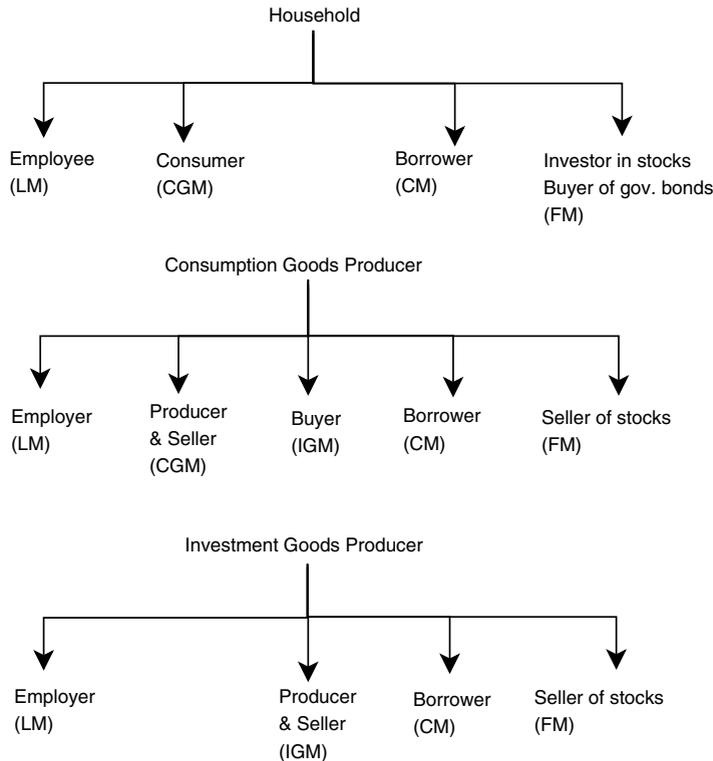
- We define the relevant list of agents.
- For each agent, we define the contexts in which this agent is supposed to act, and its respective role(s) in each context.
- For each role in a given context, we define the functions that this role should perform.

Such a hierarchy allows us to separate the functions of a single agent into several subclasses that are relevant for each distinct market context, without breaking the possible dependencies that may exist within an agent between the functions associated to different roles. All the functions of an agent can depend on all other functions of this agent, irrespective of the roles. Fig. 5 shows an example of the agent-role hierarchy for a household, and two types of firm agents: the consumption goods producers and the investment goods producers in the EURACE model.

Another advantage of our approach is that it is possible to use a similar hierarchical structure for the messages. All messages that belong to a certain market 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 market context can then be clearly delineated per context. For example, all the agents that are active on the labor market have a common subset of messages to be sent and received.

### 3.4.2. Agent activation regimes in EURACE

Since the FLAME framework uses parallel computing, all agent activities within the EURACE simulator will in principle be based on a fully asynchronous, parallel activation regime. Indeed, almost all activations in the EURACE model are event-based: the activities of an individual agent depend solely on the messages it has received and the messages it is holding



**Fig. 5.** Diagram of agent-role hierarchies for household agents, and two types of firm agents: consumption good producers and investment good producers. Each agent class is subdivided into different subclasses, reflecting the roles of the agents in different market contexts. LM: labor market, CM: credit market, FM: financial market, CGM: consumption goods market, IGM: investment goods market.

in its internal memory. That is, all the information transfer between the agents occurs through the use of messages. The messages are stored in the memory of the agent as internal variables.

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 has to retrieve and read this message before he or she can start the activity. The internal memory variables are private. It is not possible for agent A to obtain the data from agent B directly by polling B's internal memory.

Nonetheless, some activities are naturally not event-based but clock-based. Then, a central clock is being used to notify agents of the passing of time in the model, so that they can take it into account in their decision-making process. In the X-Machine framework this entails that a centralized message is broadcast to all agents, telling them that a certain date has been reached.

If agents have multiple roles (i.e. a household agent is a worker, a trader, and an investor) then one must define a meaningful procedure to activate these multiple roles within the single agent. The activation regime now becomes an intra-agent problem. After an agent is activated by the large-scale activation regime it has to determine what action to take next. To do so, the agent may use a random activation regime to chose between the different activities. Alternatively, all the agent's internal functions may get activated, implying that all the different roles of the agent are active at the same time. In an economic context, this would mean that the agent is active on different markets simultaneously. Alternatively, it is possible that certain market activities need to be completed before another market activity can take place. That is, there is an internal function dependency between the different roles of the same agent.

A second type of dependency that occurs in FLAME is the so called *communication dependency*. It occurs when a function of one agent depends on an input message sent by a different agent. All agent interactions in FLAME run through such communication dependencies, using the messages to transfer the information.

In the case of EURACE the agents are active on different markets (see Fig. 5) and at different time-scales, thereby separating their multi-role activities in time without the need for an internal time schedule to activate the different roles. In general, the roles are associated to different market contexts. The order of activation of the agents' roles in their respective market contexts should be determined by the internal function dependencies. For example, the only roles that are active in the context of the labor market are the '*worker role*' of the household and the '*employer role*' of the firm. The other roles (such as the '*consumer role*' and the '*producer role*') do not belong to the same market context. Thus, different roles of the same agent can in principle be active on different markets at the same time, acting in parallel. The interdependencies between the roles are taken into account by specifying the internal function dependencies inside each agent.

#### 4. Exploratory simulation results: the labor market

This section presents some exploratory simulations conducted with a simplified model based on EURACE's labor market module. For a detailed description of the complete model see Ref. [22].

The simplified model is composed of a capital goods sector, of a consumption goods sector with heterogenous firms, and of a labor market with heterogenous workers. The capital goods sector, however, is not agent-based, but is modeled as a passive entity whose behavior is determined by simple rules. In a nutshell, it provides an infinite supply of capital goods at exogenously given prices. The productivity of the capital goods increase over time according to a stochastic process. The amounts paid for the capital goods are channeled back into the economy.

Together with labor, the capital goods are used in the consumption goods sector to produce consumption goods. These goods are sold to the households. The firms follow plausible, to the largest possible extend empirically grounded rules for investment, production, stocking, pricing, hiring and firing, dividend payment and/or debt making. They can obtain unlimited loans at an exogenous interest rate. Likewise, the households follow plausible rules for saving, consuming different types of products, looking for a better job while employed or trying to find a job while unemployed. The contexts in which the diverse active agents are acting, their roles in the different contexts, and the main messages they send are summarized in Table 1.

The workers are characterized by (1) a general skill level and (2) specific skills. The specific skills are acquired on the job to fully exploit the technological potential of the capital used in the production process. The general skill level is obtained through schooling. The higher the worker's general skill level, the faster it acquires the specific skills associated with a given job.

Broadly speaking, the main purpose of the model is to investigate how the skill distribution in the economy influences the speed of technological change, the employment and wage dynamics, and the growth rate. Spatial aspects play a crucial role in that context, for two main reasons: (1) in most industrialized countries there are strong regional differences in the skill

**Table 1**  
Active agents, contexts, roles, and messages in the model

Agent	Context	Role	Messages
Household	Consumption goods market	Buyer	Units demanded
	Labor market	Worker	Application, accept/reject job
Firm	Investment goods market	Buyer	Units demanded
	Consumption goods market	Seller	Price, quality
	Labor market	Employer	Vacancy, job offer

distribution; and (2) geographical proximity has a crucial impact on the intensity of technological spillovers among firms, and thus, on the dynamics of technological progress.

To capture spatial effects, the economy is divided in several regions. Consumption occurs locally within each of these regions. Households and firms are distributed between the regions. Workers can apply for jobs in any region, but working outside their own region of residence is associated with commuting costs that have to be subtracted from the wage. Firms prefer to hire employees with high general skills. In case several applicants have identical general skill levels the one with the highest specific skills is selected.

#### 4.1. Labor market search-and-matching algorithm

A firm that decides to expand production posts vacancies and wage offers on the labor market to hire the additional labor needed. The firms post vacancies at most once a month. However, the labor market opens and is active everyday, since the firms are posting their vacancies asynchronously and the job seekers search for vacancies on a daily basis. The job seekers consist of the unemployed and of a randomly determined fraction  $\phi$  of employed workers who do an on-the-job search. A job seeker accepts a job only if the corresponding wage is higher than her current reservation wage. The reservation wage is a function of the worker's past wages. It decreases over time as a worker remains unemployed.

The matching algorithm between vacancies and job seekers can be summarized as follows:

- (1) Firms determine once a month their planned production output and accordingly may decide to post vacancies (indicating the wage offered and the required skills) or to fire workers.
- (2) Households look at the vacancies posted by all firms in their neighborhood, rank the vacancies according to the wages offered, and then send out job applications for those jobs which offer a wage at least equal to their reservation wage.
- (3) Firms receive the job applications, rank them according to the workers' skill-levels, and send out job offers.
- (4) Households receive job offers (possibly from multiple firms). They accept at most one job, and send rejection or acceptance messages to all firms they applied to under 2. A worker that remains unemployed lowers her reservation wage.
- (5) Firms receive the acceptance/rejection messages, and update their wage offer depending on how many vacancies are left unfilled.

The search-and-matching algorithm is illustrated in Fig. 6. It shows that the algorithm can be split up into four sequences of message sending/receiving. These steps are so called *communication layers*. They are useful to parallelize the code, since each layer can be executed independently and asynchronously from any of the other layers.

#### 4.2. Simulation results

In the simulation presented here there are two regions, with 5 firms and 200 households in each region. There are five general skill levels, 1 being the lowest and 5 the highest. The general skill levels are distributed uniformly among workers. The stochastic technology improvement process is defined such that a fully efficient economy would grow at 6% per year on average. The simulation was conducted over 4,000 periods (days), that is roughly 17 years. The values chosen for the parameters are reasonable but arbitrary – the model has not been empirically calibrated yet.

Figs. 7–9 show time series for the basic economic variables of the model: the planned and realized monthly output and actual sales, the skill-dependent unemployment rates in each of the two regions, and the average wage level. Noteworthy are the following:

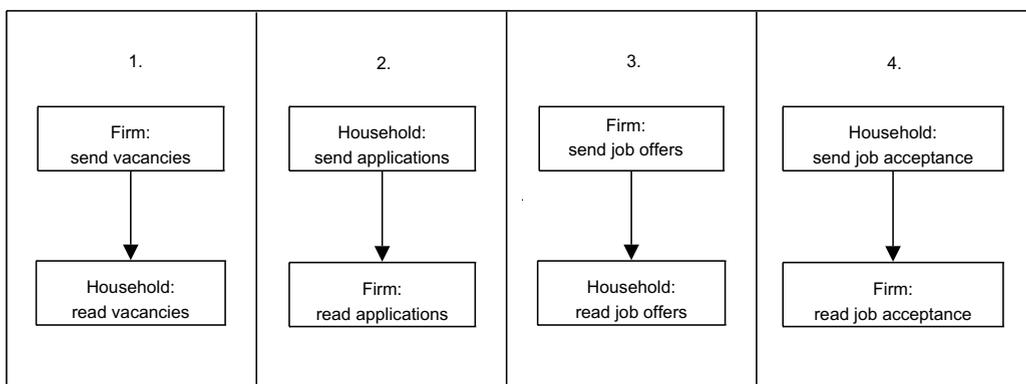


Fig. 6. Sequence of events in the labor market. The figure shows the messages that are sent between firms and job seekers in the search-and-matching algorithm.

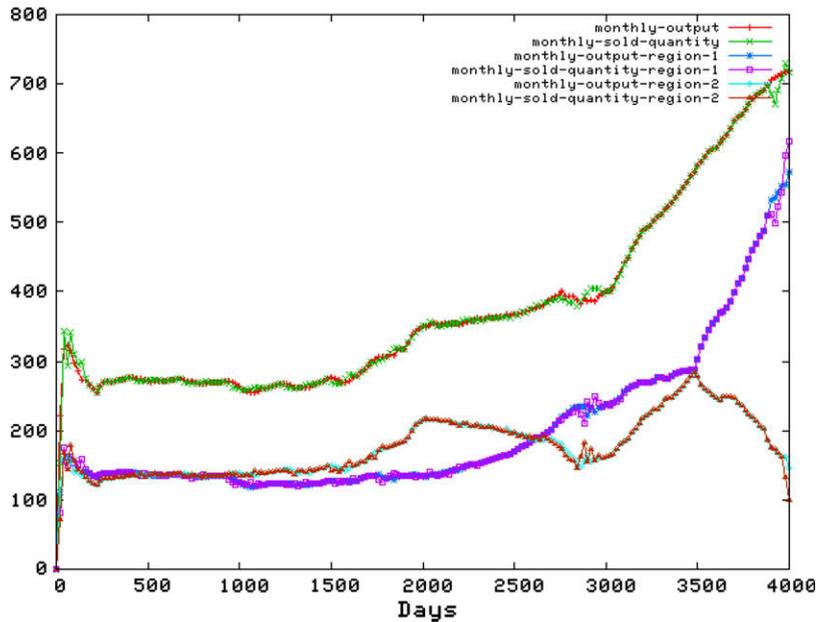


Fig. 7. Time series of monthly planned production and actual output in each region.

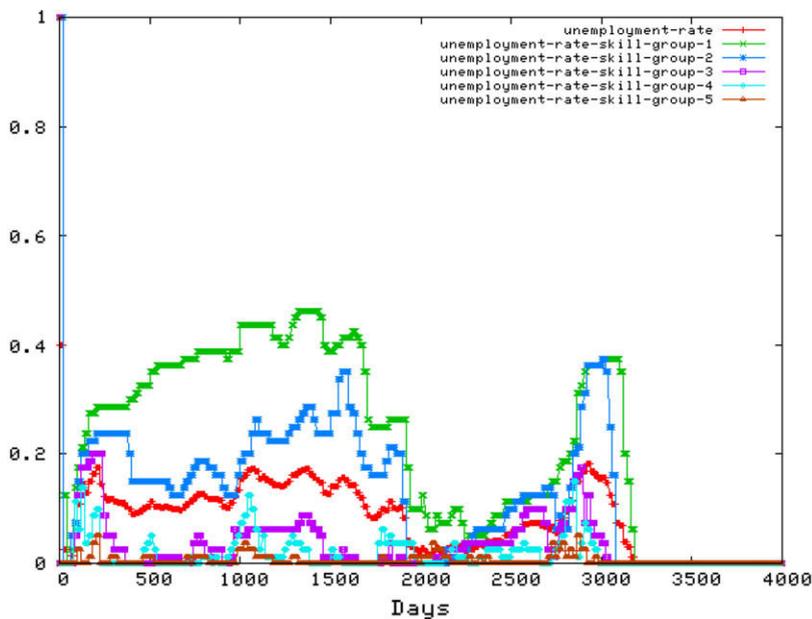


Fig. 8. Time series of the unemployment rate for each skill-level.

1. The output fluctuates over time and, after about 1000 periods, starts to markedly differ between both regions. There is a sharp increase in aggregate output starting at period 3000. However, from period 3500 on, this increase is exclusively due to output growth in region 1. The output in region 2 decreases sharply (see Fig. 7).
2. In both regions, the unemployment rates differ strongly depending on the general skill level before period 3000. The higher the skill level, the lower is the unemployment rate. After period 3000 (i.e. in the high growth phase), unemployment is almost nil for all groups. The lowest skill levels, however, are last to reach full employment (see Fig. 8).
3. The wages increase over time for all skill-levels, reflecting the exogenous increase in productivity. The speed of increase becomes much higher when the skill levels 3–5 become fully employed (around period 3200), due to the tightening of the labor market (see Fig. 9).

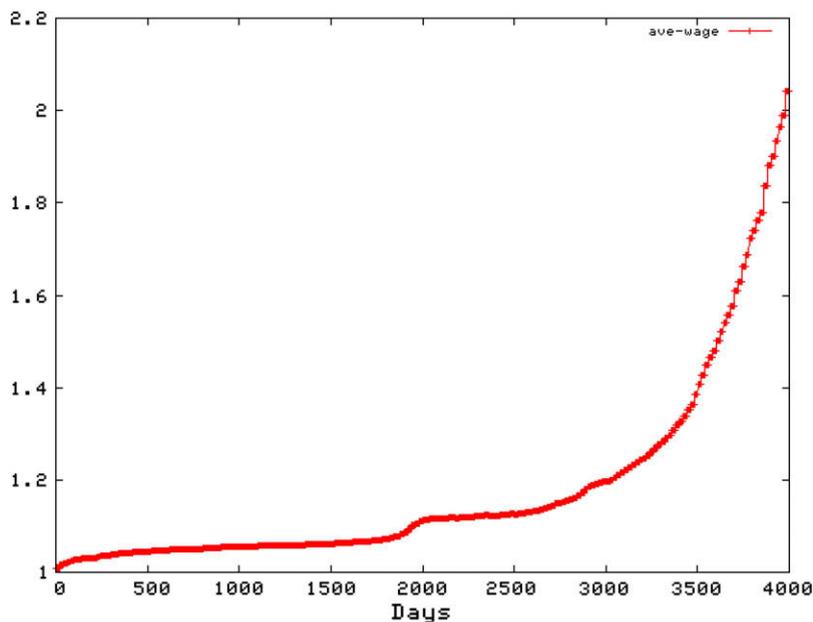


Fig. 9. Time series of the overall average wage.

- At 3% p.a. on the average, the growth rate of the economy remains much below the efficient growth rate of 6%, reflecting the inefficiency introduced by the search and matching behavior on the labor market and the local consumption goods markets.

Thus, the simplified model not only generates plausible outcomes. Starting with simple, reasonable hypotheses, it shows that even starting with almost identical initial conditions in the two regions, the emerging heterogeneity among agents may lead, after an unpredictable time, to a stark differentiation between the regional economies. It reveals that a sudden take-off can occur, again at an unpredictable time, although the (exogenous) increase in capital productivity is fairly smooth.

It gives strong insights on the inefficiencies resulting from an explicit spatial and temporal organization of markets. It provides a platform to explore, among others, the relative advantages of economic policies such as increasing the general skill level of the workers versus subsidizing the industry to increase productivity through additional R&D spending.

## 5. Conclusion

Within the EURACE project we are building an extremely large and complicated model of a much larger and more complicated real system – the European economy. It would be presumptuous to expect that the model will be able to compete, in terms of aggregate predictive power, with the standard time series and econometric models developed for decades by all main economic research centers. However, we expect to provide a number of new insights that cannot be obtained using the traditional, representative agent approaches. In particular, we are confident that our model will reproduce most of the main statistical regularities that characterize real economies and that, typically, remain unexplained and unexplainable for the economist. These are, for example, the distribution of firms' sizes, the distribution of income and wealth, diverse aspects of the spatial structure of human activities, many properties of financial time series, etc. Most importantly, the model should shed light on the way this aggregate behavior emerges from the interaction of many independent agents with much more limited cognitive and computational capabilities than classical theory presupposes. The preliminary numerical exercises we conducted suggest that we indeed are on the right way to obtain new and interesting results.

Our focus on the emergence of macro-behavior from the local, micro-interactions explains why we place so much emphasis on being able to simulate extremely large economies. This is because in a complex system many phenomena reveal themselves only when the agent population is sufficiently large. What sufficiently large concretely means, however, usually cannot be determined a priori. Thus, in addition to the technical prowess of being able to run such a large-scale parallel computational model, and to the importance of the computational techniques we are developing for many potential practical applications in other domains, our research shall shed light on the minimal size of an economy necessary for diverse self-organization phenomena to take place.

## Acknowledgements

We would like to thank our EURACE colleagues Simon Coakley and Mike Holcombe from the University of Sheffield for providing the background information on FLAME, and Chris Greenough and David Worth at the STFC Rutherford Appleton Laboratory for allowing us to use the results on the performance benchmarks.

This work made use of the facilities of HPCx, UK's national high-performance computing service, which is provided by EPCC at the University of Edinburgh and by CCLRC Daresbury Laboratory, and funded by the Office of Science and Technology through EPSRC's High End Computing Programme.

The illustrative simulations shown in the last section of this paper were carried out using a FLAME implementation of a part of the EURACE model designed and carried out by Michael Neugart, Philipp Harting, Simon Gemkov, Kordian Kabus and Klaus Wersching together with Herbert Dawid (Bielefeld University).

We gratefully acknowledge the research funding by the European Commission as part of the FP6-STREP project EURACE (under contract no. 035086).

## References

- [1] J. Arifovic, Genetic algorithm learning and the cobweb model, *J. Econ. Dyn. Control* 18 (1994) 3–28.
- [2] A.P. Kirman, N.J. Vriend, Evolving market structure: an ACE model of price dispersion and loyalty, *J. Econ. Dyn. Control* 25 (2001) 459–502.
- [3] T. Lux, M. Marchesi, Scaling and criticality in a stochastic multi-agent model of a financial market, *Nature* 397 (1999) 498–500.
- [4] B. LeBaron, Agent-based computational finance: suggested readings and early research, *J. Econ. Dyn. Control* 24 (2000) 679–702.
- [5] J. Rust, J. Miller, R. Palmer, Characterizing effective trading strategies: insights from a computerized double auction tournament, *J. Econ. Dyn. Control* 18 (1) (1994) 61–96.
- [6] H. Dawid, On the convergence of genetic learning in a double auction market, *J. Econ. Dyn. Control* 23 (9–10) (1999) 1545–1567.
- [7] H. Dawid, Agent-based models of innovation and technological change, in: L. Tesfatsion, K.L. Judd (Eds.), *Handbook of Computational Economics, Agent-based Computational Economics*, vol. 2, Elsevier, North-Holland, Amsterdam, 2006, pp. 1235–1272.
- [8] R. Axelrod, *The Evolution of Cooperation*, Basic Books, New York, 1984.
- [9] R. Axelrod, *The Complexity of Cooperation*, Princeton University Press, Princeton, 1997.
- [10] L. Tesfatsion, Agent-based computational economics: a constructive approach to economic theory, in: L. Tesfatsion, K.L. Judd (Eds.), *Handbook of Computational Economics, Agent-Based Computational Economics*, vol. 2, Elsevier, North-Holland, Amsterdam, 2006, pp. 831–880.
- [11] L. Tesfatsion (ed.), Special issue on agent-based computational economics, *Comput. Econ.* 18 (2001).
- [12] L. Tesfatsion (ed.), Special issue on agent-based computational economics, *J. Econ. Dyn. Control*, 25 (2001).
- [13] L. Tesfatsion (ed.), Special issue on the agent-based modeling of evolutionary economic systems, *IEEE T. Evolut. Comput.* 5 (2001).
- [14] H. Dawid, G. Fagiolo (Eds.), Special Issue on Agent-Based Modelling for Economic Policy Design, *J. Econ. Behav. Organ.* 67(2), (2008).
- [15] L. Tesfatsion, K.L. Judd, *Handbook of Computational Economics, Agent-Based Computational Economics*, vol. 2, Elsevier, North-Holland, Amsterdam, 2006.
- [16] S. van der Hoog, C. Deissenberg, H. Dawid, Production and Finance in EURACE, in: K. Schredelseker, F. Hauser (Eds.), *Complexity and Artificial Markets, Lecture Notes in Economics and Mathematical Systems*, Springer, 2008.
- [17] A. Gill, *Introduction to the Theory of Finite-state Machines*, McGraw-Hill, New York, 1962.
- [18] S. Coakley, *Formal Software Architecture for Agent-based Modelling in Biology*, Ph.D. Thesis, University of Sheffield, 2007.
- [19] S. Coakley, R. Smallwood, M. Holcombe, From molecules to insect communities – how formal agent-based computational modelling is uncovering new biological facts, *Sci. Math. Jpn. Online* (2006) 765–778.
- [20] G. Laycock, *The theory and practice of specification based software testing*, Ph.D. Thesis, Department of Computer Science, University of Sheffield, 1993.
- [21] S. Eilenberg, *Automata, Languages and Machines*, Vol. A, Academic Press, New York, 1974.
- [22] H. Dawid, S. Gemkov, P. Harting, K. Kabus, M. Neugart, K. Wersching, Skills, innovation, and growth: an agent-based policy analysis, *J. Econ. Stat.* (2007).