Project no.
035086
Project acronym
EURACE
Project title
An Agent-Based software platform for European economic policy design with heterogeneous interacting agents: new insights from a bottom up approach to economic modelling and simulation

Instrument: STREP

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

Deliverable reference number and title
D5.3: Software module for the agent based models of goods, labour and credit markets

Due date of deliverable:
31/08/2008
Actual submission date:

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

Organisation name of lead contractor for this deliverable
University of Sheffield - USFD

Revision 1

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

Contents

Abstract

This report presents the deliverable D5.3 accounting for the software descriptions of the models for the markets - goods and labour and credit markets. This deliverable acts as part of the work package 5 which comprises of agent based computational models of goods, labour and credit markets required for the project EURACE.

1 Introduction

2 General Model Implementation

Models are contructed from agents that have functions. These functions must be executed in a correct order for the model to run correctly. Formerly this order was defined in the model definition XMML by dependencies between functions. These dependencies could be either internal, within individual agents, or communication, dependent on messages from other agents. This was enough information to be able to order functions and plan communication sychronisation points to work in parallel.

The economic models eventually started to run only certain functions at certain times, for example, weekly or monthly. Because each function was still being executed this required a condition at the start of every function and soon became a hinderance. Even though a series of functions used the same time series they all had to include the same condition. This was also true of other conditions, for example only households that were unemployed need exectue functions involved with the labour market.

This was solved by defining the model in the XMML as a state machine. Where states are defined before and after functions. Each state can have many incoming functions and many outgoing functions. Only the unique start state has no incoming functions, and end states have no outgoing functions. This then allows branches of functions where the condition for all the functions can be defined at the start of the branch.

The model can now be recognised as a state machine but with the restriction that once a state has been entered by an agent it cannot reenter the same state. This provides sychronisation of agents in parallel during execution. Also input to functions are sets of inputs (messages) which can be empty. This is the level of detail required by FLAME to plan the communication sychronisation in parallel.

Each function can then be defined by the parameters as shown in Table 1, where M_{pre} is the pre-condition of the memory and M_{post} is the post-condition of the memory.

Current State	Input	M_{pre}	Function	M_{post}	Output	Next State

Table 1: Function parameters

In XMML this is currenly written as below where M_{pre} is defined as *condition* and M_{post} is written as source code within a function with the same name as the function name.

```
<function>
 <name>Function_name</name>
 <description>A description of the function</description>
 <currentState>current_state</currentState>
 <nextState>next_state</nextState>
 <condition>condition</condition>
 <inputs>inputs</inputs>
 <outputs>outputs</outputs>
</function>
```

3 Goods and Labour Market Implementation

4 Credit Market Implementation

5 Conclusion