REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

Agent-based modeling and simulation
Final project

Guðný Ragna Jónsdóttir, Jónheiður Ísleifsdóttir, María Arinbjarnar

— December 2005

Reykjavík University - Department of Computer Science

Technical Report

ISSN 1670-5777

REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

Agent-based modeling and simulation
Final project

Guðný Ragna Jónsdóttir, Jónheiður Ísleifsdóttir, María Arinbjarnar

Technical Report , December 2005

Abstract: Models and simulations can be very powerful tools to answer questions about complex systems like economy and human beings. Still they are not very common because of how complicated they are to build. This report will describe the design and implementation of three modules that are a part of a larger market simulation. The goal was to simulate the behavior of individuals, companies and markets in a competitive environment. Our part of the simulation was divided into three modules, the Jobcenter which regulates the employee market, the Individual which can be employed by a firm and the university which is in charge of knowledge in the system. All these modules communicate through Psyclone, a messageboard middleware. They also communicate with other modules in the system. The report also describes a monitor we built to visually observe the progress of our modules in the simulation while running. We describe the result of a simulation run and some of the major problems involved in building this kind of simulation.

Keywords: AGENT-BASED SYSTEMS; SIMULATIONS; MARKET.

(Útdráttur: næsta síða)

This project is the final project in a one semester course fall 2005 on agent-based modeling and simulations.

HÁSKÓLINN Í REYKJAVÍK
REYKJAVÍK UNIVERSITY

Lokaverkefni í
Agent-based modeling and simulation

Guðný Ragna Jónsdóttir, Jónheiður Ísleifsdóttir, María Arinbjarnar

Tækniskýrsla , Desember 2005

Útdráttur: Módel og hermilíkön geta verið mjög öflug tæki til að svara spurningum um flókna hluti eins og efnahagskerfi eða mannfólk. Þrátt fyrir þetta eru þau ekki mjög algeng þar sem þau eru flókin í smíði. Þessi skýrsla lýsir hönnun og gerð þriggja eininga sem tilheyra hermilíkani af markaði. Markmiðið var að endurspegla hegðun einstaklinga, fyrirtækja og markhópa á samkeppnis markaði. Okkar hluti af hermilíkaninu skiptist í ráðningarmiðlun sem sér um atvinnumarkaðinn, einstaklinga sem fyrirtæki geta ráðið í vinnu og háskóla sem stjórnar þekkingarþróun í líkaninu. Hlutirnir hafa samskipti sín á milli í gegnum Psyclone sem er samskiptalag sem byggir á skilaboðum. Skýrslan lýsir einnig vaktara sem við smíðuðum til að fylgjast myndrænt með framgangi eininga kerfisins á meðan á keyrslu stendur. Við lýsum niðurstöðum einnar svoleiðis keyrslu ásamt þeim vandamálum sem koma upp við smíði svona hermilíkans.

Lykilorð: HERMILÍKÖN.

(Abstract: previous page)

Contents

1 Introduction

The Constructionist approach to A.I. is a methodology that describes the use of modules to make large and complex systems. The key is that each module has a well defined interface that other modules can communicate with through predefined message types.

This project is a final project in the course Agent Based Modeling and Simulation. The aim was to simulate the behavior of individuals, companies and markets in a competitive environment.

The class was divided into 3 groups. Our group created the individual module, the jobcenter module and the university module. Our modules communicate with each other through Psyclone as well as with other modules of the simulation.

The individual can apply for a job, make it's own decisions and learn both from experience and studying at the university. There need to be several individuals in the simulation for it to work. To ease the making of several individuals we made an individual generator for creating batches of individuals.

The jobcenter is in charge of hiring the best employee for a job advertised by a firm. It chooses from applicants and lets the firm know if there are no qualified applicants for the job. It also gathers information about the supply and demand of employees and sends out information about salary accordingly.

The university handles all knowledge in the simulation. It posts information about the connection between knowledge and preference and knowledge levels at the start of the simulation. It handles the knowledge growth within the simulation, educates employees and lets them know of new knowledges that come about during the simulation.

2 Individual

The individual can apply for jobs, get hired/fired, study at the university and learn from experience. Each individual has a specific learning rate. Learning rate controls how much he learns when going to university and also how fast he will aquire greater capacity through experience.

2.1 Individual generator

This module generates individuals according to predefined specifications at the start of the simulation. The batches of individuals it generates can vary from just a few individuals to a batch of hundreds.

The generator uses input from the preferences.xml document that is used to start up the different modules of the simulation. There several variables that apply to a certain batch of

individual modules can be set and the generator then sets the variables of the individuals according to those settings and starts them.

Some of the settings are global for all batches and can only be set once for all individuals in the simulation. Those settings are:

```
<Minimum_salary>3000</Minimum_salary>
<Capacity_Increase_Factor>0.1</Capacity_Increase_Factor>
<Capacity_Decrease_Factor>0.1</Capacity_Decrease_Factor>
<Salary_change>10</Salary_change>
<Quantity_until_increase>100</Quantity_until_increase>
```

Other settings are set for each batch of individuals:

```
<batchCreate firmid="1" Count="2" initial_individualID="1" salarybottom="5000"
             salarytop="500 learningratebottom="0.6" learningratetop="0.9" >
    <knowledge id="1" levelbottom="30" leveltop="40"
          capacitybottom="10" capacitytop="30"/>
    <knowledge id="2" levelbottom="30" leveltop="40"
          capacitybottom="10" capacitytop="30"/>
</batchCreate>
```

2.2 Decisions

2.2.1 Applying for job

When the individual receives a job advertisement from a firm it checks whether it is qualified. It applies to the ad if its knowledge levels are the same or higher in at least all but one required knowledge. When applying to an advertisement the individual attaches a salary that it wants to the job application. This salary is based on what the mean salary is in the world, adjusted with some salary difference that is specific for this individual. Salary difference depends on how educated he is and how long the individual has been unemployed.

2.2.2 Being hired

When the simulation starts individuals can either start unemployed or have a job at a certain company. In the start all individuals send out a hired message to notify the firms they work at that they work there. If the individual starts out unemployed it must apply for a job at the jobcenter and if it is hired for the job it receives confirmation from the jobcenter and then informs its current employer of the hiring.

2.3 Learning

2.3.1 Learning from University

An individual's employer sends individual a command to go and study a specific knowledge at the university. The employee enrolls in the course and receives notification from the

university when the course is finished. If a new knowledge within the same preference the individual was studying has emerged, then the individual will gain this new knowledge. An individual cannot produce any products for the new knowledge before he has attended a course in it. When a knowledge level is raised the capacity for that knowledge is lowered to account for a lack of experience at that level.

2.3.2 Learning from experience

Individuals can also learn from experience. A capacity is connected to each knowledge and it stands for how many units an individual can produce in this knowledge per day. When an employed individual has been produced a certain amount in a specific knowledge it receives notification from the firm. After it has produced enough units in a specific knowledge the capacity goes up a certain percentage of the produced units.

2.4 Other functions

2.4.1 Each day

If the individual is unemployed it lowers its salary demands. The individual sends out statistics about status to the monitor.

2.4.2 Every three days

It sends out information about current status. This gives the University and Firm a chance to repeat messages that have not reached the individual.

3 Job Center

The job center handles all hiring of staff for the firms. It receives job advertisements from the firms and messages from applicants for the advertised jobs. The job center can be divided into the following subsections: Hiring appropriate individual for each job, notifying individuals about a job they have been hired for, notifying firms if no one applied to a job and calculating statistics.

3.1 Hiring

The job center saves all job advertisments in a queue as they are received. Job ads are active for a fixed number of days while applications from individuals are gathered. Applicants that do not meet the firm's expectation, either because their salary demands are too high or because they are unqualified, are filtered out. Ads from the queue are then processed in a "first in first out" manner. The job center selects a job that has some applicants for it. Each applicant's salary demands are divided with his total capacity to select a good applicant out

of those that are qualified, e.g. an experienced applicant with moderate salary demands. The chosen applicant is notified that he has been hired and the position is marked as filled.

3.2 Notifying individuals

The job center sends notification to each applicant that is successfully hired for a job, it does not send notification if the applicant was rejected for some reason. The notification contains the firm ID, the ad ID and the individual ID.

3.3 Notifying firms

The job center sends notification to the respective firm if there where no qualified applicants to a given ad within a set time limit. The message is the ad itself.

3.4 Calculating statistics

The job center calculates the following statistics and sends them to the monitor each day:

- Average salary; the weighted average salary that individuals have been hired on
- Asking salary; a combination of average salary and supply and demand.

Each day the asking salary is either incremented or decremented in respect to supply and demand of job advertisements, if there is a surplus of job ads to the demand for jobs then the asking salary is raised by a small percentage and vise versa. If an applicant is hired into a job then the weighted average is calculated into the asking salary (1). Additionally the asking salary is never incremented or decremented outside of a certain bound of the average salary, this is done so that it stays within a reasonable zone at all times.

$$\frac{(nrHired * askingSalary) + hiredSalary}{nrHired + 1} \tag{1}$$

The job center also calculates the following statistics over a 7 day period and sends them every 7 days to the monitor:

- Proportion of applicants for ads, the total number of applicants qualified or unqualified divided with all received ads for the period.
- Proportion of ads with no qualified applicants, number of sent no-applicants notification divided by the number of handled job ads for the given period.

3.5 Configurable variables

Variables that can be set for the simulation are: Number of days that are used to evaluate statistics, number of days that a job ad is active, the proportion that the asking salary is incremented or decremented by in respect to salary and demand, the corresponding bounds that the asking salary should stay inside in respect to average salary, initial asking salary, initial average salary.

```
<JobCenter name="Jobs_is"
           initial_asking_salary="5500"
           initial_average_salary="5500"
           active="True"
           asking_salary_bounds="0.3"
           asking_salary_prop="0.02"
           ad_duration="1"
           stat_duration="3"/>
```

4 University

The university handles knowledge growth in the world as well as the education of individuals.

4.1 Knowledge and Preferences

Each knowledge contains an `ID`, `current_state_of_art`, `maximum_state_of_art`, `start_day` and `preference_ID`. `start_day` is the day a knowledge will be discovered. Knowledges that are undiscovered cannot be studied or used in any way.

Each knowledge can fulfill one particular preference and as such can be used to produce products requiring that preference. Many knowledges can fulfill the same preference but no knowledge can fulfill more than one preference. Current state of art for each knowledge increases in relation to time until maximum state of art is obtained. It increases differenctly for each knowledge, growth rate is set with the `growth_speed_in_days` variable.

The firms can request a list of Preferences with associated knowledges at the start of simulation. The university only posts the Preference-Knowledge relation as it is in the beginning of simulation. Information on new knowledges within a preference cannot be attained in this way but if a knowledge for a previously unknown preference gets discovered the university posts information on that preference.

4.2 Education

The university gets a message from individual when the individual should attain a course. Individual is registered for that course for a fixed amount of time. Time unit is set with `length_in_days`.

When an individual finishes a course the university posts a message to him with a list of knowledges. The list contains the `level_increment_pr_course` of the knowledge he was studying along with any additional knowledges fulfilling the same Preference as the knowledge studied. Level of additional knowledges is set to zero. Level increment of studied knowledge is the maximum increment an individual can gain from the course. University also post an invoice to the firm employing the individual.

4.3 Statistics

Each day, the university posts the current state of art for every knowledge.

4.4 Configurable variables

Preferences.xml holds start variables for the simulation. The university part is as follows.

```
<University name="ModuleName" active="True">
    <Course price="500" length_in_days="5" level_increment_pr_course="5"/>
    <Knowledges>
        <Knowledge knowledgeID="1" start_day="0" max_state_of_art="50"
              begin_state_of_art="30"
                  growth_speed_in_days="10" preferenceID="1"/>
        <Knowledge knowledgeID="2" start_day="0" max_state_of_art="50"
         begin_state_of_art="30"
                  growth_speed_in_days="2" preferenceID="2"/>
        <Knowledge knowledgeID="3" start_day="0" max_state_of_art="150"
         begin_state_of_art="30"
                  growth_speed_in_days="1" preferenceID="3"/>
                  .
                  .
                  .
        <Knowledge knowledgeID="10" start_day="100" max_state_of_art="750"
         begin_state_of_art="30"
                  growth_speed_in_days="9" preferenceID="5"/>
    </Knowledges>
</University>
```

5 Monitor

The monitor uses JFreeGraph to graphically display status of various elements throughout the simulation. Graphic is live and charts are updated each simulation day. The monitor is categorized by Job Center, Individual, University and Messages. The monitor does not send any messages or communicate with any modules in the system except as a receiver of various statistical messages.

For each individual there is a panel with 3 line charts as shown in figure 1. We show individual's salary if he is working and his asking salary when he is unemployed. While an individual is employed this is actually the same figure since an individual always gets the salary he asks for. We also show the individual´s knowledge, and how that changes through time, one chart for the levels of each knowledge and one chart for the production capacity of each knowledge. We also show various states of the individual such as learning rate and status. Status of all individual is also shown in a single piechart divided into Employed, Studying or Unemployed.

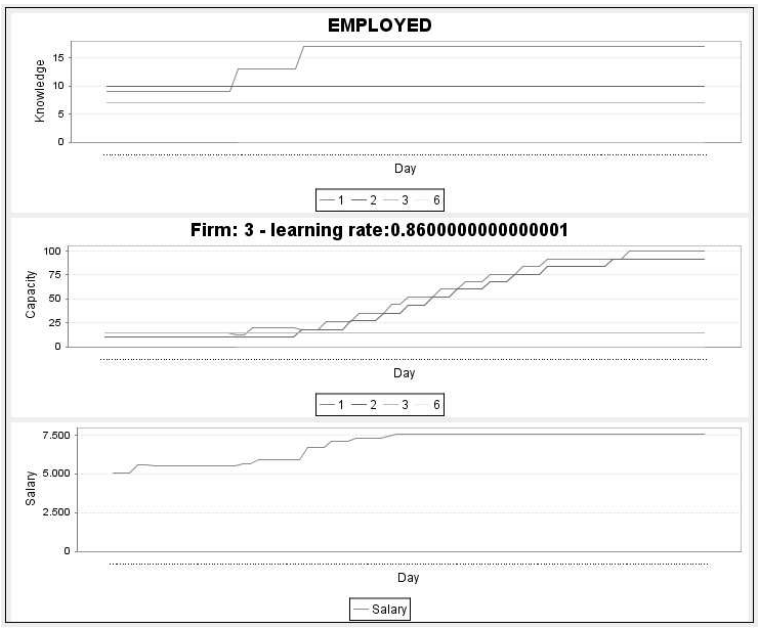

Figure 1: Monitoring Individual.

From the University the monitor shows a line chart of current state of art of each knowledge in the simulation, shown in figure 2. Knowledges that are undiscovered according to the startup settings are not shown until their start day comes up.

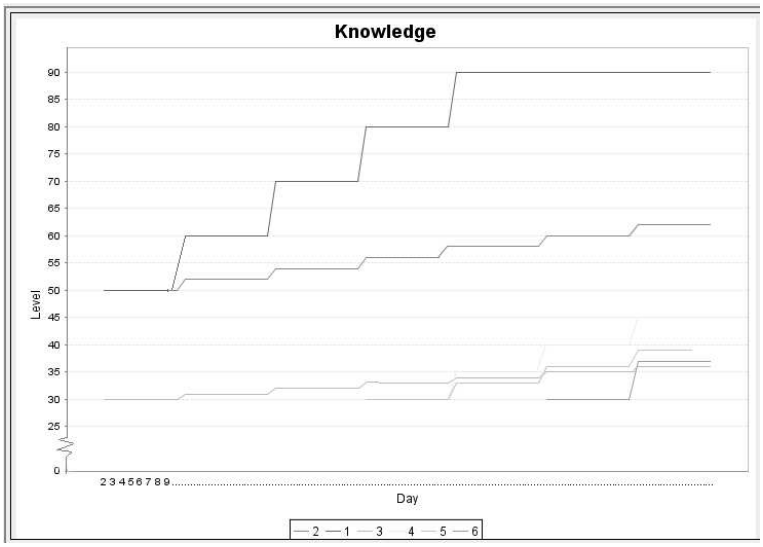

Figure 2: Monitoring University.

From the Job Center the monitor shows a line chart for salaries. Trend in actual average of all employed individuals is shown as well as development in asking salary for the individuals. This is shown in figure 3.

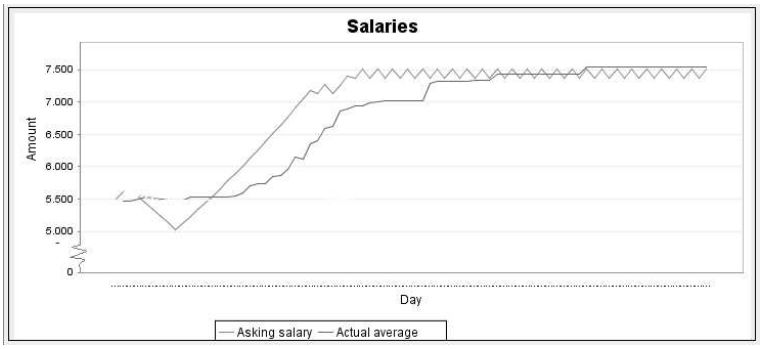

Figure 3: Monitoring Job center.

6 Time

We kept a fairly detailed log of the time spent on the project. The time does only include time spent on the model itself and it does not include time spent on this report.

Work	Gudny	Jonheidur	Maria	Total
Defining the whole versus individual units	4	4	4	12
University module	35	0	0	35
JobCenter module	2	0	44	46
Individual module	12	30	0	42
Monitor	25	0	0	25
Getting two units to work together	5	5	7	17
Getting more then two units to work together	22	10	15	47
Getting all the units to work together	10	6	8	24
Run the system before all the units are ready	20	10	30	60
Defining message types	2	2	1	5
Defining a base for the system	25	2	2	29
The whole system runs for over three minutes	10	8	6	24
Bugs, bug fixes	10	10	12	32
Psyclone, IntelliJ	5	2	1	8
Redesigning a unit	3	3	2,5	8,5
Splitting a unit	0	0	0	0
Merging two or more units	0	0	0	0
Architecture	4	4	4	12
Changing the architecture	5	5	5	15
Communicating with other groups	8	5	5	18
	207	106	146,5	459,5

7 Results

For a typical run of the simulation we included 30 individuals, 3 companies, 2 target groups, university, job center and a bank. Xml document for startup preferences is located in Appendix.

Firms start by hiring people and continue to try to do so until they are filling every production order. Companies fill a bigger part of the orders each day until they are completely filled. After that happenes a status quo is sustained.

When simulation was run with few employees we always got the same behaviour. All individuals are hired very early in the simulation and when no individuals are answering employment ads, the companies start educating their current employees. At this point there is a huge demand for employees but no supply. This leads to wages climbing. Wages keep climbing until at some point the companies feel that they cannot afford the employees and

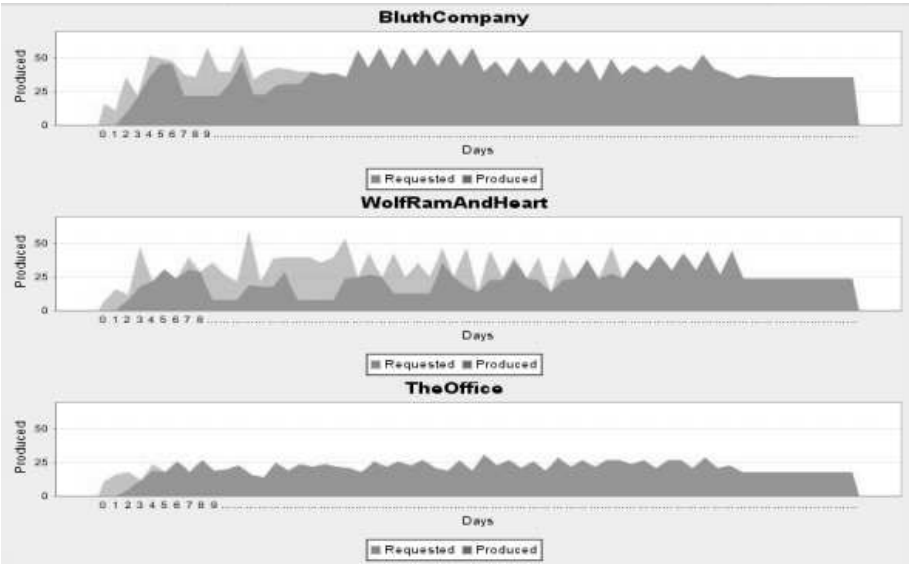

Figure 4: Production stabilized

employees are let go. This causes the supply/demand of workers to even out and wages to start dropping to normal. There is also a built in desperation of unemployed individuals and they tend to lower their demands much quicker than the average salary takes to drop. This means that it does not take such a long time for the companies to be able to hire somebody within their wage range. The run then repeats this behaviour again and again.

8 Discussion

During test runs we had repeated problems with messages being so delayed that they did not reach the intended agent in time for a normal functionality to take place. We tried numerous ways to solve this problem and finally managed to get a fairly good run by splitting Psyclone between 4 computers and having the rest of the simulation run on the 5'th computer. This problem was made additionally difficult by the difficulties in synchronising the clocks on the computers. We needed to greatly reduce the number of messages and to increase the span of the day to be able to get a reasonable test run from the simulation. This time synchronizing problem is a well known problem of distributed system and is frequently solved by defining a logical time and by marking messages with timestamps and implementing some time synchronising functionality.

This project demanded on one hand skills of abstraction and co-operation and on the other programming skills. Firstly to be able to abstract sufficiently without loosing sight of important details that need to be consistent through out the implementation, for instance

time and message types. Secondly to be able to code the resulting high level design in detail in a sufficiently generic and object oriented way to ensure that the individual modules will be able to interact with each other and deal with all kinds of unexpected erroneous behaviour.

9 Conclusion

We have shown that the Individual, Job centre and University all function to the specified requirements. The Individual manages to apply for a job, work at a company, get education and make calculated decisions about his salary demands in respect to his level of education and desperation for a job. The Job centre receives job ads and hires a suitable applicant for the job if any is available. The University educates any individual that is sent to it and informs individuals of new knowledges within the same preference as the knowledge studied. All of these modules also inform the monitor of any relevant statistics and information so that their successful processing can be monitored.

10 Appendix

10.1 Startup preferences

```
<?xml version="1.0" encoding="utf-8" ?>
<!--
Setup 09.11.2005 in Room 431, revision 456 of classes
Psyclone running on machines: 5525 and whiteboards on 5461, 5459, 5458
Market: 5471, Firms: 5465, Individuals 5477
Results: Seems to work properly with a 20 sec day, distributed on above
    machines.
-->
<StartValues>
    <Start_of_simulation>15:34:00</Start_of_simulation>
    <Server>localhost</Server>
    <Port>10000</Port>
    <Length_of_day>20000</Length_of_day>

    <FirmMonitor name="FirmMonitor" active="True"/>
    <IndividualMonitor name="IndividualMonitor" active="True"/>
    <MarketMonitor name="MarketMonitor" active="True"/>

    <University name="RU" active="True">
        <calculate_level_in_days>10</calculate_level_in_days>
        <Course price="500" length\_in_days="5" level_increment_pr_course="5"/>
        <Knowledges>
            <Knowledge knowledgeID="1" start_day="0" max_state_of_art="90"
                       begin_state_of_art="50" growth_speed_in_days="10"
                       preferenceID="1"/>
            <Knowledge knowledgeID="2" start_day="0" max_state_of_art="120"
              begin_state_of_art="50" growth_speed_in_days="2"
              preferenceID="2"/>
            <Knowledge knowledgeID="3" start_day="0" max_state_of_art="150"
              begin_state_of_art="30" growth_speed_in_days="1"
              preferenceID="3"/>
            <Knowledge knowledgeID="4" start_day="20" max_state_of_art="100"
              begin_state_of_art="30" growth_speed_in_days="5"
              preferenceID="4"/>
            <Knowledge knowledgeID="5" start_day="20" max_state_of_art="250"
              begin_state_of_art="30" growth_speed_in_days="3"
              preferenceID="5"/>
            <Knowledge knowledgeID="6" start_day="40" max_state_of_art="250"
              begin_state_of_art="30" growth_speed_in_days="7"
```

```
          preferenceID="1"/>
        <Knowledge knowledgeID="7" start_day="100" max_state_of_art="350"
          begin_state_of_art="30" growth_speed_in_days="8"
          preferenceID="2"/>
        <Knowledge knowledgeID="8" start_day="100" max_state_of_art="350"
          begin_state_of_art="30" growth_speed_in_days="1"
          preferenceID="3"/>
        <Knowledge knowledgeID="9" start_day="100" max_state_of_art="650"
            begin_state_of_art="30" growth_speed_in_days="3"
            preferenceID="4"/>
        <Knowledge knowledgeID="10" start_day="100" max_state_of_art="750"
            begin_state_of_art="30" growth_speed_in_days="9"
            preferenceID="5"/>
    </Knowledges>
</University>

<JobCenter name="Jobs_is" initial_asking_salary="5500"
        initial_average_salary="5500" active="True"
        asking_salary_bounds="0.3" asking_salary_prop="0.02"
        ad_duration="1" stat_duration="3"/>

<IndividualStatistics name="Individuals" active="True"/>

<Individuals active="True">
    <Minimum_salary>3000</Minimum_salary>
    <Capacity_Increase_Factor>0.1</Capacity_Increase_Factor>
    <Capacity_Decrease_Factor>0.1</Capacity_Decrease_Factor>
    <Salary_change>10</Salary_change>
    <Quantity_until_increase>100</Quantity_until_increase>
    <batchCreate firmid="1" Count="1" initial_individualID="21"
     salarybottom="5000" salarytop="6000"
             learningratebottom="0.3" learningratetop="0.9" >
        <knowledge id="1" levelbottom="6" leveltop="10"
         capacitybottom="5" capacitytop="10"/>
        <knowledge id="2" levelbottom="6" leveltop="10"
         capacitybottom="5" capacitytop="10"/>
    </batchCreate>
    <batchCreate firmid="2" Count="1"
       salarybottom="5000" salarytop="6000"
              learningratebottom="0.3" learningratetop="0.9">
        <knowledge id="1" levelbottom="6" leveltop="10"
         capacitybottom="5" capacitytop="10"/>
```

```
          <knowledge id="2" levelbottom="6" leveltop="10"
           capacitybottom="5" capacitytop="10"/>
      </batchCreate>
      <batchCreate firmid="3" Count="1"
       salarybottom="5000" salarytop="6000"
                learningratebottom="0.3" learningratetop="0.9">
          <knowledge id="1" levelbottom="6" leveltop="10"
           capacitybottom="5" capacitytop="10"/>
          <knowledge id="2" levelbottom="6" leveltop="10"
           capacitybottom="5" capacitytop="10"/>
      </batchCreate>
      <batchCreate firmid="0" Count="7"
       salarybottom="5000" salarytop="6000"
                learningratebottom="0.3" learningratetop="0.9">
          <knowledge id="1" levelbottom="6" leveltop="10"
           capacitybottom="10" capacitytop="15"/>
          <knowledge id="2" levelbottom="6" leveltop="10"
           capacitybottom="10" capacitytop="15"/>
          <knowledge id="3" levelbottom="6" leveltop="10"
           capacitybottom="10" capacitytop="15"/>
      </batchCreate>
  </Individuals>

<Firms>
   <!-- This first firm shows the default settings of the firm and which
        policies will be used if any of the policy tags are omitted.
   -->
   <Firm name="BluthCompany" firmid="1" capital="5000000"
    maxSalary="10000" explore="false" active="True">
      <Policies>
         <EmploymentPolicy value="HIRE_LAST"/>
         <EmploymentIdleTolerancePolicy value="MEDIUM"/>
         <KnowledgeIdleTolerancePolicy value="MEDIUM"/>
         <AlertnessPolicy value="MEDIUM_LOW"/>
         <EducationPolicy value="TRAIN_IF_NOT_HIRE"/>
         <EducationSelectionPolicy value="TRAIN_BEST_EMPLOYEE"/>
         <ProductAssignmentPolicy value="DONT_CARE"/>
         <!--ADD DISABLE EXPLORING SWITCH-->
         <!--ADD PRICE ADJUSTMENT THRESHOLD-->
         <!--ADD TOGGLE PRICE ADJUSTMENT-->
      </Policies>
      <Product price="10000.0">
```

```
      <PreferenceRequests>
         <PreferenceRequest preferenceID="1" requestedLevel="5"/>
         <PreferenceRequest preferenceID="2" requestedLevel="5"/>
      </PreferenceRequests>
   </Product>
</Firm>
<Firm name="TheOffice" firmid="2" capital="5000000"
 maxSalary="10000" explore="false" active="True">
   <Policies>
      <EmploymentPolicy value="HIRE_LAST"/>
      <EmploymentIdleTolerancePolicy value="MEDIUM"/>
      <KnowledgeIdleTolerancePolicy value="MEDIUM"/>
      <AlertnessPolicy value="MEDIUM_HIGH"/>
      <EducationPolicy value="TRAIN_IF_NOT_HIRE"/>
      <EducationSelectionPolicy value="TRAIN_BEST_EMPLOYEE"/>
      <ProductAssignmentPolicy value="DONT_CARE"/>
   </Policies>
   <Product price="10000.0">
      <PreferenceRequests>
         <PreferenceRequest preferenceID="1" requestedLevel="5"/>
         <PreferenceRequest preferenceID="2" requestedLevel="5"/>
      </PreferenceRequests>
   </Product>
</Firm>
<Firm name="WolfRamAndHeart" firmid="3" capital="5000000"
 maxSalary="10000" explore="false" active="True">
   <Policies>
      <EmploymentPolicy value="HIRE_LAST"/>
      <EmploymentIdleTolerancePolicy value="MEDIUM"/>
      <KnowledgeIdleTolerancePolicy value="MEDIUM"/>
      <AlertnessPolicy value="HIGH"/>
      <EducationPolicy value="TRAIN_IF_NOT_HIRE"/>
      <EducationSelectionPolicy value="TRAIN_BEST_EMPLOYEE"/>
      <ProductAssignmentPolicy value="DONT_CARE"/>
   </Policies>
   <Product price="10000.0">
      <PreferenceRequests>
         <PreferenceRequest preferenceID="1" requestedLevel="5"/>
         <PreferenceRequest preferenceID="2" requestedLevel="5"/>
      </PreferenceRequests>
   </Product>
</Firm>
```

```
</Firms>
 <!-- Possible ProductShift Policy Values:
      BetterProduct - the group shifts toward a higher preference
       products without knowing if they exist or not.
      WorseProduct - the group shifts toward a lower preference
       products without knowing if they exist or not.
      NeareasatBetterProducts - the group shifts towards the nearest
        advertised better product.
      BestBetterProduct - the group shifts toward the best advertised
       product.
  -->

 <Bank name="Bank" Capital="1000000000.0" rate="5.0" active="True"/>

 <TargetGroups >
     <TargetGroup name="TGA" id="1" activateAfter="100" quantity="150"
      maxprice="15000" numberofinterrupts="3" maxdistance="4"
      active="True">
         <Policies>
             <PriceShiftPolicy averagetime="15" variance="5" active="True" />
             <ProductShiftPolicy value="BetterProduct" averagetime="15"
              variance="5" maxProductDistance="10" active="True" />
             <SizeShiftPolicy averagetime="15" variance="5" active="True" />
         </Policies>
         <PreferenceRequests>
             <PreferenceRequest preferenceID="1" requestedLevel="5"/>
             <PreferenceRequest preferenceID="2" requestedLevel="5"/>
         </PreferenceRequests>
     </TargetGroup>
     <TargetGroup name="TGB" id="2" activateAfter="100" quantity="140"
         maxprice="25000" numberofinterrupts="3" maxdistance="8"
         active="True">
         <Policies>
             <PriceShiftPolicy averagetime="10" variance="2" active="True" />
             <ProductShiftPolicy value="BetterProduct" averagetime="3"
                  variance="2" maxProductDistance="15" active="False" />
             <SizeShiftPolicy averagetime="10" variance="2" active="True" />
         </Policies>
         <PreferenceRequests>
             <PreferenceRequest preferenceID="1" requestedLevel="13"/>
             <PreferenceRequest preferenceID="2" requestedLevel="13"/>
         </PreferenceRequests>
```

```
        </TargetGroup>
    </TargetGroups>
</StartValues>
```

10.2 Posters and Triggers

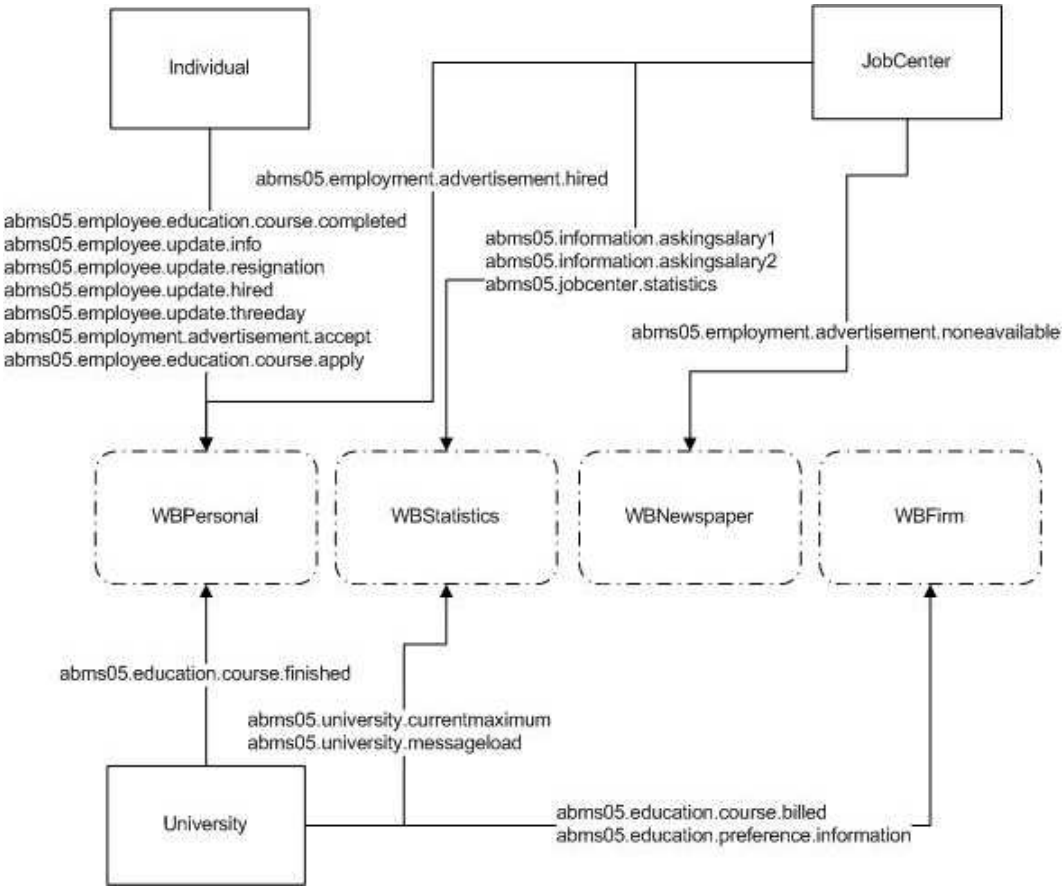

Figure 5: Messages posted by each module.

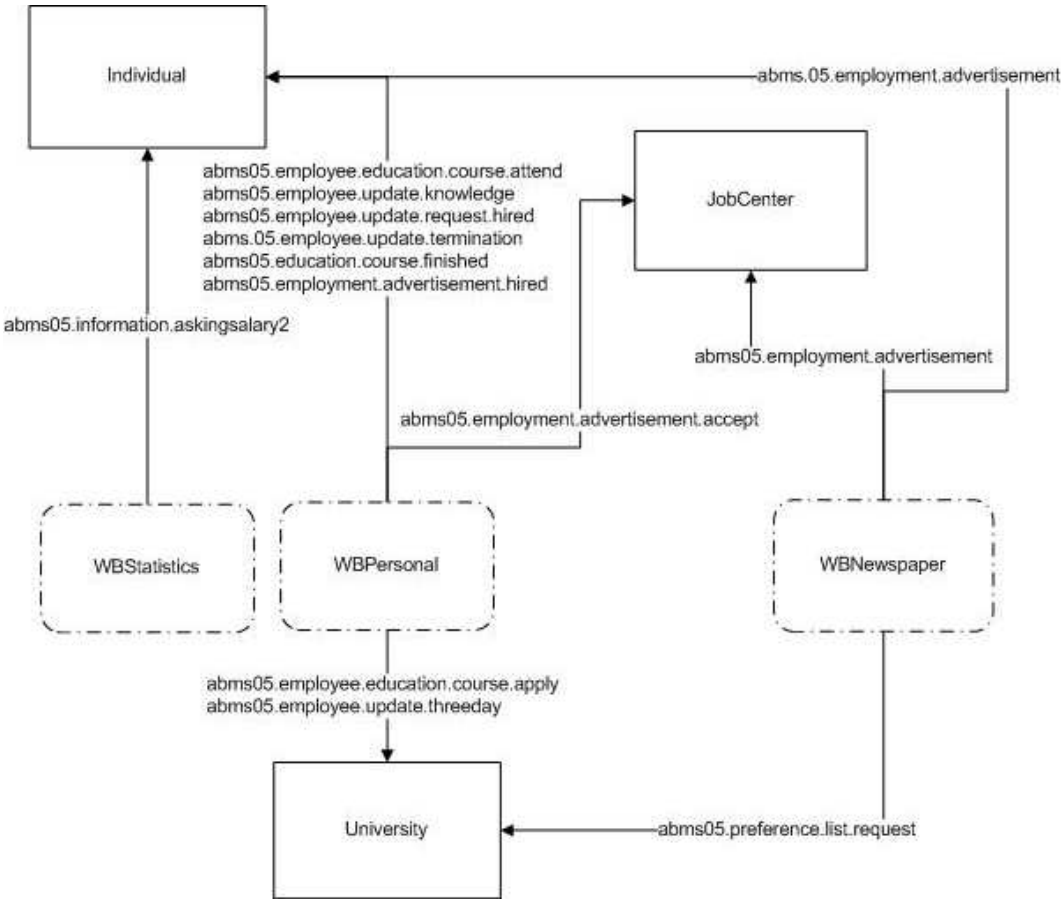

Figure 6: Messages each module subscribes to.

REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

Department of Computer Science
Reykjavík University
Ofanleiti 2, IS-103 Reykjavík, Iceland
Tel: +354 599 6200
Fax: +354 599 6201
http://www.ru.is
ISSN 1670-5777