

ИСПОЛЬЗОВАНИЕ МНОГОАГЕНТНЫХ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ СОСТАВЛЕНИЯ РАСПИСАНИЙ

**И.В. Крысина (Нижегородский филиал государственного университета –
Высшей школы экономики)**

**Научный руководитель – к.т.н., доцент Э.А. Бабкин (Нижегородский филиал
государственного университета – Высшей школы экономики)**

В рамках работы была предпринята попытка разработки агентного алгоритма составления расписания для вуза. Была разработана программная реализация алгоритма, проведен анализ эффективности работы алгоритма, его сравнение с результатами работы человека.

Введение

Невозможно переоценить роль расписания времени в жизни современного общества. На основе расписаний осуществляется планирование деятельности как каждого отдельного человека, так и огромных промышленных комплексов. Это доказывает, что одним из самых важных навыков современного человека становится умение правильно распределять свое рабочее время. Идеальное расписание обязательно должно обладать следующим свойством: все пользователи получают максимально возможную выгоду от его использования. В связи с этим возникает необходимость учета требований, предъявляемых пользователями к расписанию, и нахождения оптимального расписания при заданных ограничениях. Т.е. решается задача максимизации полезности каждого индивида. Задача усложняется, если пользователи удалены друг от друга: возникает проблема обмена информацией и фиксирования достигнутых соглашений

В связи с этим встает вопрос: нельзя ли заменить пользователей расписания программными агентами, способными обмениваться сообщениями друг с другом с целью поиска наилучшего решения. Тогда рутинный процесс совместного составления расписания можно будет заменить автоматизированным процессом взаимодействия агентов-представителей в рамках некоторого программного комплекса.

Подобные программные комплексы представляются очень перспективными в свете роста актуальности проблемы составления расписаний. В частности, к ним можно отнести задачи составления расписаний (движение поездов и самолетов, занятия учебных групп в вузе, выполнение вычислительных задач на ВЦ коллективного пользования или в многопроцессорной компьютерной системе, операции на станке с программным управлением, задания в годовом плане предприятия, передача пакетов данных в сети Интернет).

Целью данного исследования является разработка агентного алгоритма составления расписания учебного заведения.

Научная новизна исследования заключается в применении агентного подхода к решению задачи составления учебного расписания для вуза. Ранее при решении данной проблемы применялись алгоритмы линейного программирования [3], методы поиска с использованием метаэвристик [5], имитация остывания [2, 6], кластерные методы [7].

При составлении расписания учитываются предпочтения пользователей о времени и месте проведения занятий. Это свойство предложенного алгоритма позволяет повысить качество получаемого расписания. Принципиальное отличие предложенного алгоритма от ранее разработанных заключается в том, что он позволяет распределять ресурсы аудиторного фонда между пользователями расписания и, в принципе, любые другие виды ресурсов. Другие агентные алгоритмы составляют только расписание времени [4]. Большинство современных систем составления расписания учебных заведений не позволяют пользователям напрямую влиять на процесс составления расписания [1]. Практическая значимость данного исследования состоит в том, что предложен спо-

соб обработки предпочтений пользователей, что, в свою очередь, повышает качество получаемых расписаний.

Важной характеристикой любой системы составления расписаний является способность быстро реагировать на изменения входных параметров. Агентный алгоритм в случае изменения предпочтений пользователей расписания изменит лишь ту часть расписания, которая имеет отношение к данному пользователю. Таким образом, исключается повторное решение задачи.

В следующих разделах статьи приведены математическая постановка задачи, описание многоагентного алгоритма и описание архитектуры разработанного приложения.

Математическая постановка задачи [8]

В целях упрощения изложения лиц, заинтересованных в результатах процесса составления расписания, будем называть пользователями расписания. К пользователям расписания относятся преподаватели и учебные группы. Роль первых – провести лекционное или практическое занятие, роль последних – присутствовать на занятиях. Введем следующие обозначения.

Учебные группы и потоки. $g \in \mathbf{G}$ – номер учебной группы. \mathbf{G} – множество номеров всех учебных групп. $|\mathbf{G}| = \gamma$ – количество групп.

Каждая из групп принадлежит одному или нескольким потокам. При объединении нескольких групп в один поток используются следующие принципы.

1) Группы в потоке используют один и тот же аудиторный фонд. Например, поток «Гуманитарные предметы, ИСиТ, 4-й курс», состоящий из групп 04-ПМ, 04-БИ, использует аудиторный фонд из двух аудиторий на 120 мест.

2) Лекции читаются всему потоку сразу.

3) У каждого потока есть хотя бы одно занятие.

4) Поток может состоять из одной учебной группы. Например, поток «Прикладная математика, предметы по специальности, 4-ый курс» состоит из одной группы 04-ПМ.

\mathbf{R} – множество номеров учебных потоков. $|\mathbf{R}| = \rho$ – количество потоков. $r \in \mathbf{R}$ – номер потока. Каждая группа является самостоятельным потоком, поэтому $\rho \geq \gamma$.

$\mathbf{C}_r \subset \mathbf{G}$ – поток. Поток называется любое подмножество множества групп. $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_\rho\}$ – множество всех потоков. Потоки могут пересекаться между собой.

Преподаватели. $p \in \mathbf{P}$ – номер преподавателя, \mathbf{P} – множество всех преподавателей. $|\mathbf{P}| = \pi$ – количество преподавателей.

Пользователи расписания. Объединение множества всех групп с множеством всех преподавателей $\mathbf{M} = \mathbf{G} \cup \mathbf{P}$, $m \in \mathbf{M}$ – уникальный номер пользователя расписания.

Время. $w \in \mathbf{W}_g$ – номер учебного дня недели. $\mathbf{W}_g \subset \mathbf{W} = \{1, 2, \dots, 7\}$ – множество учебных дней группы g , \mathbf{W} – множество всех дней недели. $j \in \mathbf{J} = \{1, 2, \dots, 8\}$ – номер учебной пары. $\mathbf{T} = \{(w, j) | w \in \mathbf{W}, j \in \mathbf{J}\}$ – множество таймслотов (элементарных единиц времени в задаче составления расписания). Например, таймслот (1,2) означает понедельник, вторая пара. Для каждого пользователя m известно множество таймслотов, когда он свободен, т.е. в это время занятие может быть проведено $\mathbf{T}_m^+ \subset \mathbf{T}$, и множество недоступных таймслотов $\mathbf{T}_m^- \subset \mathbf{T}$ ($\mathbf{T}_m^+ \cup \mathbf{T}_m^- = \mathbf{T}$; $\mathbf{T}_m^+ \cap \mathbf{T}_m^- = \emptyset$), например, выходные дни, когда занятия не могут быть проведены.

Занятия. Преподаватели проводят лекционные и практические занятия. Лекционные занятия проводятся у всего потока сразу, практические занятия – только у одной

группы. Кроме того, практические занятия требуют другого аудиторного фонда. Например, практические занятия по информатике должны проходить в компьютерном классе. Введем следующие обозначения для занятий.

$S_r = \{1, 2, \dots, \sigma_r\}$ – множество номеров лекционных занятий, читаемых на потоке r . $s_r \in S_r$ – номер лекционного занятия;

$Q_r = \{1, 2, \dots, \theta_r\}$ – множество номеров практических занятий, проводимых на потоке r . $q_r \in Q_r$ – номер практического занятия.

Можно видеть, что глобально уникальным идентификатором занятия будет пара «номер потока – номер занятия на потоке». Действительно, занятия существенно зависят от потока. Так, «физика на 2-м курсе» отличается по содержанию от «физика на 3-м курсе».

Лекционное занятие однозначно идентифицируется парой значений «номер потока – номер лекционного занятия»: $(r, s_r) \in \mathbf{RS}$, где

$$\mathbf{RS} = \{(r, s_r) \mid r \in \mathbf{R}, s_r \in S_r\} \quad (0.1)$$

есть множество всех лекционных занятий.

Количество лекционных занятий

$$|\mathbf{RS}| = \sum_{r=1}^p \sigma_r. \quad (0.2)$$

Практическое занятие однозначно идентифицируется тройкой значений «номер потока – номер практического занятия – номер группы»: $(r, g_r, q_r) \in \mathbf{RQG}$, где

$$\mathbf{RQG} = \{(r, g_r, q_r) \mid r \in \mathbf{R}, q_r \in Q_r, g_r \in C_r\} \quad (0.3)$$

есть множество всех практических занятий.

Количество практических занятий

$$|\mathbf{RQG}| = \sum_{r=1}^p |C_r| \cdot \theta_r,$$

где $|C_r|$ – количество групп в потоке C_r .

При дальнейшем описании задачи нам в большинстве случаев будет неважно, какое занятие рассматривается – лекционное или практическое. Поэтому под занятием будем понимать элемент из объединения двух множеств – множества лекционных и множества практических занятий:

$$\mathbf{E} = \mathbf{RS} \cup \mathbf{RQG}. \quad (0.4)$$

Учебный план закрепляет за каждым преподавателем предметы, которые он должен будет провести в течение семестра. Назначение лекционных занятий:

$$\delta_1 : \mathbf{RS} \rightarrow \mathbf{P},$$

где \mathbf{P} – множество преподавателей; \mathbf{RS} – множество лекционных занятий.

Учебный план практических занятий:

$$\delta_2 : \mathbf{RQG} \rightarrow \mathbf{P},$$

где \mathbf{RQG} – множество троек «номер потока – номер практического занятия – номер группы». Например, $C_1 = \{3, 4, 5\}$ – поток 1 состоит из групп 3, 4, 5; $Q_1 = \{1, 2\}$ – у потока 1 всего два практических занятия; запись $\delta_2(1, 2, 4) = 7$ означает, что у группы с номером 4, которая входит в поток 1, практическое занятие номер 2 ведет преподаватель 7.

В общем случае учебный план можно записать так:

$$\delta : \mathbf{E} \rightarrow \mathbf{P}, \quad (0.5)$$

$$\text{где } \delta(e) = \begin{cases} \delta_1(e), e \in \mathbf{RS} \\ \delta_2(e), e \in \mathbf{RQG} \end{cases}.$$

Зная учебный план, можно вычислить множество занятий, проводимых пользователем-преподавателем (или для пользователя-группы) с номером m :

$$\begin{aligned} \mathbf{E}_m &= \{e \mid m \in \mathbf{P} \wedge \delta(e) = m\} \cup \\ &\{e = (r, s) \mid m \in \mathbf{C}_r \wedge s \in \mathbf{S}_r\} \cup \\ &\{e = (r, q, m) \mid m \in \mathbf{C}_r \wedge q \in \mathbf{Q}_r\} \end{aligned} \quad (0.6)$$

Аудиторный фонд. Аудиторный фонд представляет собой множество аудиторий, где могут быть проведены занятия.

\mathbf{A} – множество всех аудиторий и помещений.

Для каждого занятия e выделяется некоторое подмножество аудиторий $\mathbf{A}_e \subset \mathbf{A}$; аудитория для проведения занятия выбирается только из этого подмножества.

Неизвестные функции задачи. Неизвестными в данной задаче являются, расписание времени и расписание аудиторий. Расписание времени – это отображение множества предметов на множество таймслотов:

$$\tau : \mathbf{E} \rightarrow \mathbf{T}, \quad (0.7)$$

где \mathbf{E} – множество всех занятий; \mathbf{T} – множество всех таймслотов.

Расписание аудиторий – это отображение множества занятий на множество аудиторий:

$$\alpha : \mathbf{E} \rightarrow \mathbf{A}, \quad (0.8)$$

где \mathbf{E} – множество всех занятий; \mathbf{A} – множество всех аудиторий.

Пример 1. $\alpha(1,2) = 101$ означает, что занятие 2 потока 1 проводится в аудитории 101.

Пример 2. $\mathbf{C}_1 = \{3,4,5\}$ – поток 1 состоит из групп 3, 4, 5; $\mathbf{Q}_1 = \{1,2\}$ – у потока 1 всего два практических занятия; запись $\alpha(1,2,4) = 101$ означает, что у группы с номером 4, которая входит в поток 1, практическое занятие номер 2 будет проводиться в аудитории 101.

Ограничения. Введем в задаче следующие ограничения.

1. Один преподаватель в каждый момент времени может проводить не более одного занятия.

$$\forall p \in \mathbf{P}, \forall e_1, e_2 \in \mathbf{E} : e_1 \neq e_2 \wedge \delta(e_1) = \delta(e_2) = p \Rightarrow \tau(e_1) \neq \tau(e_2) \quad (0.9)$$

2. В одной аудитории в каждый момент времени может проводиться не более одного занятия.

$$\forall a \in \mathbf{A}, \forall e_1, e_2 \in \mathbf{E} : e_1 \neq e_2 \wedge \alpha(e_1) = \alpha(e_2) = a \Rightarrow \tau(e_1) \neq \tau(e_2) \quad (0.10)$$

3. У одной группы в каждый момент времени может проводиться не более одного занятия.

$$\begin{aligned} \forall g \in \mathbf{G} : (e_1 = (r_1, \dots), e_2 = (r_2, \dots) \in \mathbf{E} \wedge g \in \mathbf{Cr}_1 \wedge g \in \mathbf{Cr}_2 \wedge e_1 \neq e_2) \vee \\ \vee (e_1 = (\dots, g), e_2 = (\dots, g) \in \mathbf{E} \wedge e_1 \neq e_2) \Rightarrow \tau(e_1) \neq \tau(e_2) \end{aligned} \quad (0.11)$$

Другими словами, для каждых двух пересекающихся потоков в каждый момент времени не может быть разных лекционных занятий в одно и то же время, и для каждой группы не может быть разных практических занятий в одно и то же время. Это ограничение удобно проверять, предварительно разбив все занятия на γ классов. В класс $\mathbf{K}_g \subset \mathbf{E}, g \in \mathbf{G} (g_1 \neq g_2 \Rightarrow \mathbf{K}_{g_1} \cap \mathbf{K}_{g_2} = \emptyset)$ входят все занятия, которые читаются группе

g персонально или в каком-либо потоке. Ограничение 3 будет выполнено тогда и только тогда, когда все занятия групп одного класса проводятся в разное время.

Приоритеты занятий. Очевидно, что не все занятия имеют одинаковую важность в рамках учебного процесса. Для них можно задать порядок по приоритету. Более приоритетные занятия будут в первую очередь занимать время и место, менее приоритетные занятия уже не смогут претендовать на эти ресурсы. Итак, приоритеты занятий представляют собой отношение частичного порядка на множестве занятий

$$e_1 \succ e_2 \Leftrightarrow U(e_1) \geq U(e_2), \quad (0.12)$$

где $e_1, e_2 \in \mathbf{E}$; $U(e) = |\mathbf{M}_e| + k_1(e) + k_2(e) + k_3(p_e)$ – «полезность» занятия e .

$\mathbf{M}_e = \{m \mid (e = (r, s) \in \mathbf{RS} \wedge m \in \mathbf{C}_r) \vee (e = (r, q, m) \in \mathbf{RQG})\}$ – множество групп, у которых проводится занятие e ;

$k_1(e) \in \{0, 5, 10\}$ характеризует степень важности занятия для специальности (0 – «ненужные занятия», 5 – важное, но не по специальности, 10 – занятие по специальности);

$k_2(e) \in \{0, 2\}$ – 0 – младшие курсы, 2 – старшие курсы;

$p_e = p, \delta(e) = p$ – номер преподавателя, который ведет занятие e ;

$k_3(p_e) \in \{0..5\}$ – оценка преподавателя как научного сотрудника; более высокий коэффициент означает, что преподаватель – более ценный научный сотрудник. В случае равенства $U(e_1) = U(e_2)$ занятия упорядочиваются в лексикографическом порядке.

Предпочтения пользователей. Важной частью предлагаемой модели являются предпочтения пользователей. Каждое предпочтение представляет собой числовую оценку в диапазоне от 0 до 1. 0 соответствует наименьшему уровню предпочтения, 1 – наибольшему. В модели имеют место два вида предпочтений:

1. предпочтения пользователя $m \in \mathbf{M}$ о времени проведения занятия:

$$f_m^1 : \mathbf{E}_m \times \mathbf{T}_m^+ \rightarrow [0, 1] \quad (0.13)$$

2. предпочтения пользователя $m \in \mathbf{M}$ о месте проведения занятия:

$$f_m^2 : \mathbf{EA}_m \rightarrow [0, 1], \quad (0.14)$$

где $\mathbf{EA}_m = \{(e, a) \mid a \in \mathbf{A}_e \wedge e \in \mathbf{E}_m\}$ – множество допустимых пар «занятие – аудитория».

Графически предпочтения можно представить в виде табл. 1 и 2. Более темный цвет отмечает более предпочтительное время (место) для проведения занятия.

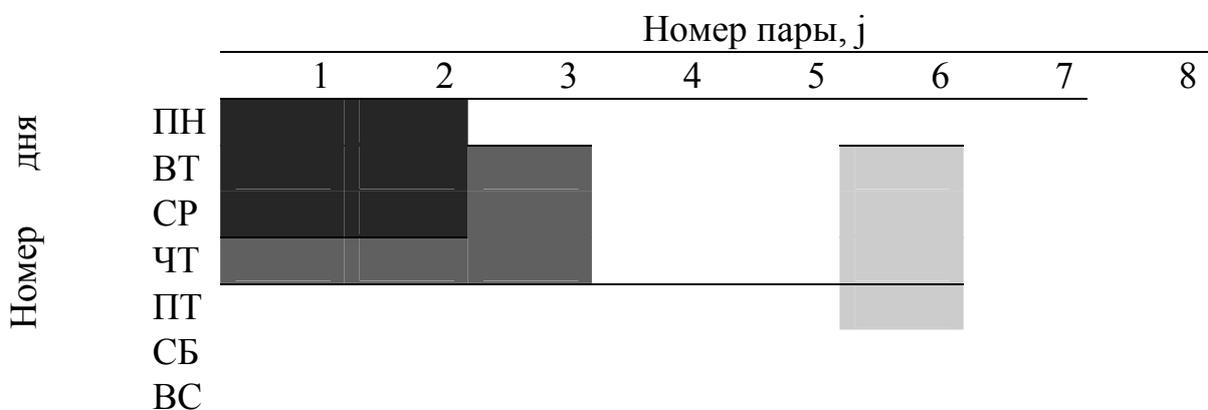


Таблица 1. Предпочтения пользователя m о времени проведения занятия, f_m^1

Аудитория,	1	2	3	4	5
а					
Оценка, f_m^2					

Таблица 2. Предпочтения пользователя m о месте проведения занятия, f_m^2

Критерий качества решения. Критерий качества представляет собой обобщенный критерий, состоящий из нескольких частных критериев. Критерий оценивает найденное решение, т.е. пару функций $\tau(e), \alpha(e)$.

Частный критерий

$$F_e^1(\tau) = \sum_{m \in M_e} f_m^1(e, \tau(e)) \rightarrow \max \quad (0.15)$$

определяет сумму предпочтений пользователей о времени занятия e . Частный критерий

$$F_e^2(\alpha) = \sum_{m \in M_e} f_m^2(e, \alpha(e)) \rightarrow \max \quad (0.16)$$

определяет сумму предпочтений пользователей о месте проведения занятия e . Обобщенный критерий

$$F(\tau, \alpha) = \sum_{e \in E} (F_e^1 + F_e^2) \rightarrow \max \quad (0.17)$$

вначале максимизирует пару частных критериев для наиболее приоритетного занятия, затем для следующего по приоритету и т.д. При этом в паре критериев вначале максимизируется первый критерий (время), затем второй (место). Такая структура обобщенного критерия гарантирует, что более приоритетные занятия получают лучшее время и место. Считается, что время проведения занятия более важно при составлении расписания, поэтому вначале определяется время, а затем место проведения занятия.

Решение. Решением задачи являются неизвестные функции τ, α при условии выполнения всех ограничений и максимальном значении критерия качества.

Описание алгоритма

В разделе дается описание алгоритма составления расписания учебного заведения. Помимо расписания времени, составляется еще и расписание аудиторий.

Агенты разделяются на агентов преподавателей, агентов учебных групп и агентов аудиторий. Роли организаторов выполняют агенты преподавателей, роли участников выполняют агенты учебных групп и аудиторий. Количество агентов каждого типа определяется количеством соответствующих пользователей. Агент аудиторий – один для каждого расписания.

Поиск времени и места для проведения занятия осуществляется посредством деятельности агента преподавателя. Данный агент способен отправлять и получать различные типы сообщений, переходя при этом из одного состояния в другое. Ниже приводится описание действий, выполняемых в каждом состоянии.

1) Ask_when_avail. Первое состояние, в котором оказывается агент преподавателя. В этом состоянии он посылает агентам групп запрос WHENAVAIL с номером занятия, чтобы они сообщили свое свободное время. Агенты групп отвечают сообщением USERAVAIL, в котором сообщается, когда агент свободен. Если все агенты сообщили свой ответ, тогда агент преподавателя находит пересечение по времени. Если пересечение пусто, тогда агент переходит в состояние (1a) imposs_meeting.

1a) *Imposs_meeting*. Состояние, в которое агент попадает, если свободное время агентов групп данного занятия имеет пустое пересечение, т.е. решения найти не удалось. Составить расписание для занятия не удалось. Занятие помечается «не имеющее решения».

2) *Ask_subj_prefs*. В этом состоянии агент преподавателя опрашивает агентов групп, чтобы они сообщили свои предпочтения о времени и месте проведения занятия (сообщение *EVALUATE*). Агенты сообщают свои предпочтения в сообщении *SUBJPREFS*. В этом сообщении содержатся предпочтения о времени и предпочтения о месте проведения занятия. Полученные предпочтения времени сортируются в соответствии с критерием алгоритма *MSRAC*, то есть по возрастанию приоритета. Полученные предпочтения о месте тоже сортируются по убыванию предпочтений, т.е. первая аудитория будет наиболее предпочтительной.

3) *Propose_time*. Агент выбирает первый таймслот из полученного на предыдущем шаге упорядочения времени. Агент устанавливает это время в сообщении и посылает запрос агентам учебных групп (*TIMEPROPOSAL*), содержащий номер занятия и предлагаемый таймслот. Те проверяют свой календарь. Если в предложенное время никаких занятий нет, тогда агент группы дает положительный ответ – *ACCEPT*. Если же в календаре уже есть другое занятие на это время, тогда агент группы сравнивает приоритеты нового и старого занятий. Если приоритет нового занятия больше, то агент группы соглашается (*ACCEPT*), иначе отказывается (*REJECT*). В случае равных приоритетов используется критерий метрополиса. В случае согласия агент преподавателя переходит в следующее состояние, в случае отказа агент возвращается в состояние *propose_time* и выбирает следующий таймслот. Если таймслотов больше нет, значит, решения для данного занятия не существует. Агент переходит в состояние (3a) *solnot_found*.

3a) *Solnot_found*. Делается пометка, что для занятия расписания составить не удалось.

4) *Propose_location*. Агент посылает отсортированный список аудиторий, полученный в состоянии (2) агенту аудиторий. *LOCPROPOSAL* содержит номер занятия, номер таймслота и список возможных аудиторий, отсортированных по предпочтению. Агент аудиторий получает *LOCPROPOSAL* и, используя свой календарь «аудитория – время – занятие», находит первую аудиторию из списка, которая в указанно время либо не занята, либо занята менее приоритетными занятиями. Если аудитория найдена, то агент аудиторий посылает ответ *ACCEPT* с номером аудитории, иначе агент посылает ответ *REJECT*. При получении положительного ответа агент преподавателя переходит в следующее состояние. Если же был получен отрицательный ответ, то агент преподавателя возвращается в состояние *propose_time* для определения нового времени занятия.

5) *Fix_meeting*. Если агент преподавателя приходит в это состояние, значит, время и место были успешно найдены. Агент преподавателя рассылает всем агентам групп и агенту аудиторий сообщение *FIXMEETING* с номером занятия, номером таймслота и номером аудитории. Если у агентов групп в календаре в указанное время пусто и у агента аудиторий указанное место пусто, то время и место занятия фиксируются. Иначе, если например, время или место уже заняты более приоритетным занятием, то менее приоритетное занятие отменяется с помощью сообщения *CANCEL MEETING*, отправляемого агенту преподавателя отменяемого занятия. Преподаватель, получив это сообщение, затем пересылает его всем участникам. Участники убирают указанное занятие из своего календаря.

В состояниях (1), (2), (3) и (4), если не все агенты прислали ответ в течение некоторого времени, то агент преподавателя помещает занятие в список отмененных занятий, потом он попытается заново найти для него расписание.

После того, как все агенты преподавателей завершат поиск расписания своих занятий, общее расписание считается составленным. Важным является тот факт, что частичное расписание всегда будет составлено. Полное расписание всех занятий иногда просто не существует, в таком случае рассмотренный алгоритм все равно найдет расписание, просто менее приоритетные занятия будут автоматически отброшены.

Описание архитектуры приложения

При разработке приложения использовались следующие программные продукты и технологии.

- Java – в качестве основного языка программирования. Выбор был обусловлен необходимостью обеспечения возможности переноса приложения с одной платформы на другую, гибкостью и простотой разработки графического интерфейса, возможностью запуска приложения на мобильных платформах (сотовые телефоны, коммуникаторы, КПК).
- MySQL – СУБД. Критерии выбора: свободное распространение, высокая скорость обработки запросов, поддержка транзакций, поддержка JDBC.
- JADE (Java Agent DEvelopment Framework) – многоагентная система, разрабатываемая Итальянским институтом телекоммуникаций при поддержке таких компаний, как Motorola, France Telecom R&D. Критерии выбора – свободное распространение, открытость, полное соответствие стандартам (FIPA), возможность исполнения на мобильных устройствах.

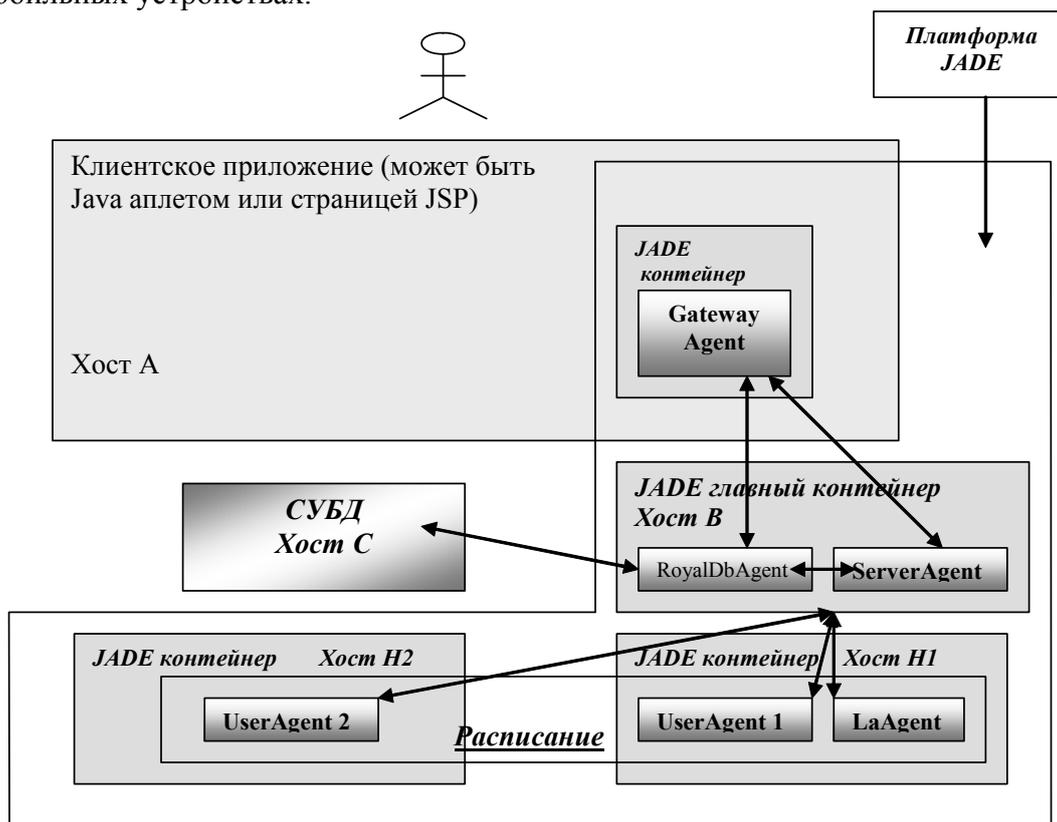


Рисунок. Состав системы и связи между компонентами

Архитектура разрабатываемого ПО составления расписания в некоторой степени зависит от архитектуры используемой многоагентной системы. Прикладная программа, использующая JADE, имеет дело со следующими объектами.

1. Агент. Программная сущность, способная исполняться в рамках системы JADE.
2. Сообщение. Агенты обмениваются информацией друг с другом и с агентной системой с помощью сообщений. Любое взаимодействие основано только на сообщениях.
3. Контейнер. Среда для исполнения нескольких агентов одновременно. Агенты загружаются в контейнер и затем запускаются. На одном компьютере может быть запущено несколько контейнеров одновременно. Один контейнер на нескольких компьютерах запущен быть не может.
4. Платформа. Является логическим объединением нескольких контейнеров. При старте платформы автоматически запускается главный контейнер (main container). Платформа поддерживает службу директорий, которая позволяет находить агентов, используя шаблон для поиска. Агенты разных платформ могут общаться друг с другом.
Состав системы и связи между компонентами условно изображены на рисунке.

Заключение

В рамках данной работы был разработан агентный алгоритм составления расписания для вуза и представлена его программная реализация.

- 1) Разработан агентный алгоритм составления учебного расписания. Разработанный алгоритм, помимо расписания времени, позволяет составлять еще и расписание аудиторий.
- 2) Алгоритм дает хорошее приближение к оптимальному решению, т.е. получаемое на выходе расписание хорошо отражает предпочтения пользователей о времени и месте проведения занятий в смысле критериев исходной задачи.
- 3) Разработан программный комплекс, который демонстрирует работу алгоритма на практике.
- 4) Измеренное на практике время работы алгоритма хорошо согласуется с теоретическими оценками вычислительной трудоемкости.
- 5) Эффективность работы алгоритма сопоставима с эффективностью работы человека.

Возможно продолжение исследований в данной области с целью расширения сферы применения алгоритма – например, создание на его основе многоагентной системы планирования маршрутов общественного транспорта.

Литература

1. Севастьянов С.В. Введение в теорию расписаний // Новосибирск: Новосибирский государственный университет. – 2003. – 171 с.
2. Abramson D. and Dang H. School Timetables: A Case Study in Simulated Annealing. Applied Simulated Annealing: Lecture Notes in Economics and Mathematics Systems // Springer-Verlag. – 1993. – V. 2. – P. 89–93.
3. Akkoyunly E.A. A Linear Algorithm for Computing the Optimum University Timetable // The Computer Journal. – 1973. – № 16(4). – P. 347–350.
4. BenHassine A., D'efago X., Ho T.B. Agent-Based Approach to Dynamic Meeting Scheduling Problems // Proceeding to Int. Conf. AAMAS-04. – 2004. – V. 3. – №. 1. – P. 1132–1139.
5. Petrovic S., Burke E. University Timetabling. // University of Nottingham. – 2003.
6. Shaerf A. Local Search Techniques for Large High School Timetabling Problems // IEEE Transactions on Systems, Man And Cybernetics. – 1999. – V. 29. – № 4. – P. 368–377.
7. White G.M., Chan P.W. Towards the Construction of Optimal Examination Timetables // INFOR. – 1979. – V. 17. – P. 219–229.
8. Ретинский И.М. Проектирование и реализация алгоритмов составления расписания на основе многоагентных технологий: Дис. ... магистер. – Нижегородский государственный технический университет, 2005. – 97 с.