

УДК 004.75, 004.62, 004.942

МОДЕЛИРОВАНИЕ ВЗАИМОДЕЙСТВИЯ АГЕНТОВ В МНОГОАГЕНТНОЙ СИСТЕМЕ С ПОМОЩЬЮ ЦВЕТНЫХ СЕТЕЙ ПЕТРИ И НЕЧЕТКОЙ ЛОГИКИ

А.В. Владимиров, аспирант

*(Вологодский государственный технический университет,
ул. Ленина, 15, г. Вологда, 160000, Россия, iclever7@yandex.ru)*

Выделены проблемы в сфере функционирования крупных адаптивных информационных систем, а также основные требования, что позволило сформировать структуру адаптивной информационной системы и определить ее основные сегменты. Приведены функции компонентов системы и описан механизм ее работы. Выделена структура агентной составляющей адаптивной информационной системы с рассмотрением функций агентов. С использованием аппарата цветных сетей Петри смоделирован процесс взаимодействия агентов в многоагентной адаптивной информационной системе. Приведены основные множества цветов, выделены основные составляющие модели, исследована работа модели на разных этапах и произведен анализ основных свойств системы. Описаны процесс обработки задач агентами системы, основные состояния агентов, используемые в процессе обработки задач, а также механизмы очередей при работе системы. Приведено описание агентов в виде кортежа. Рассмотрена модель оценки качества построения. На основе полученного общего набора продукционных правил, оптимизированного для повышения устойчивости и адекватности модели, а также определенных нечетких лингвистических переменных построен прототип системы нечеткого вывода. Рассмотрены терм-множества лингвистических переменных, и сделана оценка качества построения на основе входных данных. Проанализировано влияние изменения входных нечетких лингвистических переменных на выходную переменную. Получена и проанализирована трехмерная визуализация модели оценки качества.

Ключевые слова: многоагентные системы, адаптивные системы, моделирование взаимодействия агентов, цветные сети Петри, агентная архитектура, информационные системы, агентно-ориентированная архитектура, нечеткая логика, системы нечеткого вывода.

MODELING AGENTS INTERACTION IN A MULTI-AGENT SYSTEM USING THE APPARATUS OF COLOURED PETRI NETS AND FUZZY LOGIC

Vladimirov A.V., Postgraduate Student

(Vologda State Technical University, Lenina St., 15, Vologda, 160000, Russian Federation, iclever7@yandex.ru)

Abstract. The article identifies problems in functioning of major adaptive information systems, as well as the basic requirements. It allows forming a structure of an adaptive information system and highlighting its main segments. The paper also shows functions of system components and describes its action. A structure of agent-based component of the adaptive information system with the consideration of the functions of agents is obtained. Using the apparatus of colored Petri nets the author models the process of agents' interaction in a multi-agent adaptive information system. A basic set of colors is shown, the main model components are considered, model work at different stages is researched, an analysis of the main systems properties is made. The article describes tasks processing by systems agents, basic states of agents used in processing tasks, queues mechanisms in system functioning. A description of the agents is shown like a tuple. A model of construction quality evaluation is considered. A prototype of fuzzy inference system is built based on obtained common set of production rules, optimized to enhance the sustainability and adequacy of the model, as well as certain fuzzy linguistic variables. A term set of linguistic variables are considered. The construction quality estimation based on input data is made. The effect of variations of input fuzzy linguistic variables for the output variable is analyzed. The three-dimensional visualization of the quality assessment model is obtained and analyzed.

Keywords: multiagent systems, adaptive systems, modeling agents' interactions, coloured Petri nets, agent-based architecture, information systems, agent-oriented architecture, fuzzy logic, fuzzy inference system.

В настоящее время необходимы системы, способные адаптироваться под возникающие к их функционированию требования. В некоторых случаях нужно адаптировать работу системы при минимальном вмешательстве человека или без такового вообще. В сфере функционирования крупных адаптивных информационных систем важной является проблема внутреннего взаимодействия компонентов, обусловленная наличием больших объемов неупорядоченной разнородной информации.

Адаптивные информационные системы позволяют динамически учитывать предоставляемые

персональные требования пользователей, перераспределять рабочие процессы между компонентами системы, реагировать на возникающие проблемные ситуации, более эффективно использовать ресурсы системы.

Обозначим алгоритм взаимодействия информационной системы с пользователем, а также ее структуру (рис. 1). Выделим три сегмента в системе: административный и сервис-агентный сегмент, интеллектуальный сегмент, сегмент оперирования данными.

Административный и сервис-агентный сегмент включает в себя инструменты администрирования

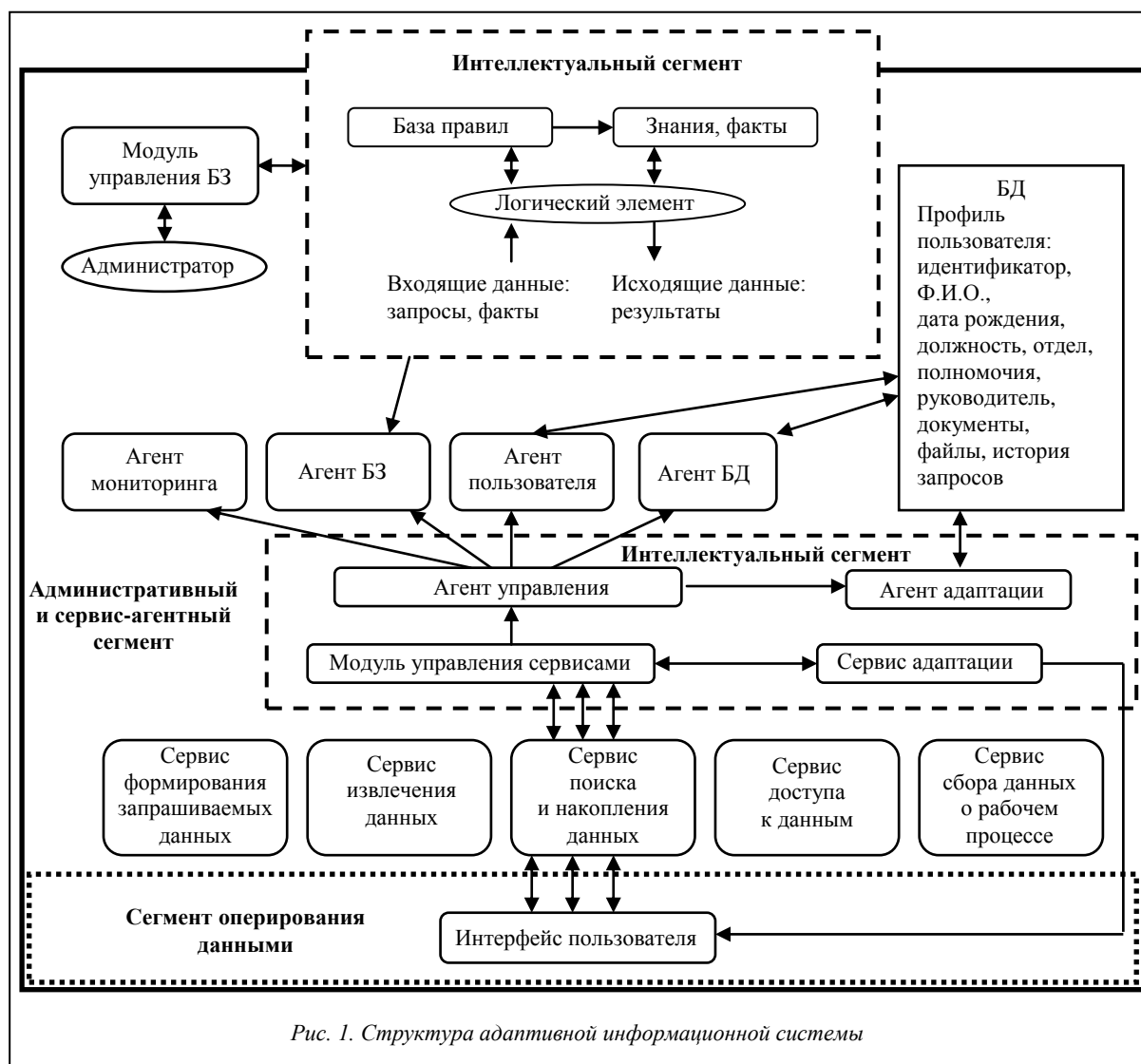


Рис. 1. Структура адаптивной информационной системы

БЗ, БД, агента мониторинга, агента БЗ, агента пользователя, агента БД, сервис формирования запрашиваемых данных, сервис извлечения данных, сервис поиска и накопления данных, сервис доступа к данным, сервис сбора данных о рабочем процессе. Интеллектуальный сегмент включает БЗ, агента управления, модуль управления сервисами, сервис адаптации, агента адаптации, а сегмент оперирования данными – интерфейс пользователя.

В процессе работы системы от пользователя поступает воздействие с последующим запуском соответствующих процессов по обработке данных. В ситуации, связанной с принятием определенного рода решения и получением информации из БЗ, осуществляется взаимодействие с интеллектуальным сегментом. Выполняются вызов соответствующего сервиса с передачей данных, описывающих текущее состояние системы, а также запрос, который должна выполнить интеллектуальная подсистема. Механизм логического вывода информации реализуется с помощью разноцелевых

сервисов и агентов. Поиск решения сервисом или агентом основывается на входных данных и информации, хранящейся в БЗ, которая модифицируется экспертом через модуль управления БЗ и включает в себя базы правил, факты, информацию, описывающую интерпретацию фактов. Данный модуль также выполняет такие функции, как контроль непротиворечивости системы правил и контроль правильности ввода новых знаний [1].

При рассмотрении структуры адаптивной информационной системы стоит уделить внимание сегменту, содержащему набор разноцелевых агентов, а также моделированию их взаимодействия. Далее более детально разберем агентную составляющую информационной системы, включающую агентов-исполнителей: агента мониторинга, агента БЗ, агента пользователя, агента БД, агента адаптации, а также агента управления. Агент управления осуществляет координацию работы подчиненных ему агентов, постановку задач. Среди функций агентов-исполнителей можно выделить сбор необходимой информации и ее подготовку. Агент

мониторинга следит за текущим состоянием агентов в системе и количеством поступивших задач. Агент БД осуществляет взаимодействие с БД и резервное копирование информации. Агент БЗ передает входные запросы к БЗ. Агент пользователя позволяет собирать информацию о действиях пользователей. Агент адаптации собирает данные о функционировании используемых пользователем компонентов системы [1].

Процесс взаимодействия между агентами в многоагентной информационной системе рассмотрим с использованием аппарата цветных сетей Петри. Моделирование позволяет описывать и анализировать процессы, протекающие в системе.

Рассмотрим функционирование агентной составляющей с использованием аппарата цветных сетей Петри, в которых важное место занимает типизация данных, основанная на использовании множества цветов. Граф сетей Петри состоит из позиций и переходов. Каждый переход имеет такие атрибуты, как условие запуска, время задержки и выполняемое действие, а каждая позиция – начальную маркировку и тип позиции. В позициях сети размещаются фишки с определенным типом данных [2].

Рассмотрим построение агентной составляющей и ее моделирование в среде CPN Tools на следующем примере. На рисунке 2 изображена сеть Петри, моделирующая процесс на стадии распределения задач.

В данной ситуации определено множество цветов *tasks* с элементами *tasks_for_database_agent*, *tasks_for_adaptation_agent*, *tasks_for_knowledgebase_agent*, *tasks_for_monitoring_agent*, *tasks_for_user_agent*, *task_for_agent_coordination*, обозначающими соответствующие задачи для каждого из имеющихся агентов. Позиция *coordination service* имеет множество цветов *q* с фишкой *coordination_task*, переходящей в позицию *agent coordination*, тем самым инициализируется процесс координации задач агентом управления. Переменные *a, b, c, d, e, f, i, h, x, z* используются для извлечения фишек из входных позиций и помещения новой фишки в выходную позицию. Агент координации распределяет задачи в соответствии с указанным типом задачи. Агент мониторинга с использованием счетчиков Counter собирает информацию о количестве задач у каждого из агентов с целью выявления степени загруженности определенных агентов. Если текущая задача находится в процессе обработки, то следующая задача добавляется в конец очереди Queue. Задачи обслуживаются по принципу: «первым пришел – первым обслужен» (FIFO). В случае отсутствия очереди незамедлительно начинают обрабатываться новые поступающие задачи.

Рассмотрим модель промежуточной стадии моделирования (рис. 3). Некоторое количество задач ожидают обработки агентами. Агент БЗ, агент

пользователя, агент адаптации выполняют обработку поступившей задачи. Наибольшее количество задач выполняется агентом координации.

В процессе обработки задач агентами используются следующие состояния: *knowledgebase agent free* (агент БЗ свободен), *knowledgebase agent busy* (агент БЗ занят), *knowledgebase agent use* (агент БЗ используется), *user agent free* (агент пользователя свободен), *user agent busy* (агент пользователя занят), *user agent use* (агент пользователя используется), *database agent free* (агент БД свободен), *database agent busy* (агент БД занят), *database agent use* (агент БД используется), *adaptation agent free* (агент адаптации свободен), *adaptation agent busy* (агент адаптации занят), *adaptation agent use* (агент адаптации используется). Также используются состояние *wait* (ожидание обработки агентом) и состояние *timer 2c* (таймер на 2с) типа Boolean, которое служит для моделирования задержки, принимая от перехода захвата агента фишку с временным штампом. Фишки, находящиеся при инициализации обработки задачи в позициях, свидетельствующих о том, что агент свободен, в ходе работы сети отражают его состояние (свободен или занят), при этом цвет фишки содержит информацию о задаче, обрабатываемой агентом. После поступления задачи на обработку переход, свидетельствующий о том, что агент захвачен, сработать не сможет, так как нет фишки в позиции, свидетельствующей о том, что агент свободен. Не разрешен также переход, свидетельствующий о том, что агент выполняет обработку из-за того, что фишка, имитирующая таймер, несет временной штамп, равный 2000. Если установлено, что нет доступных переходов, происходит наращивание счетчика глобальных часов до тех пор, пока не откроется переход, свидетельствующий о том, что агент выполняет обработку, так как штамп фишки в позиции *timer 2c* окажется равным значению глобальных часов. В результате начнется освобождение агентов посредством срабатывания перехода, свидетельствующего о том, что агент выполняет обработку. Таким образом, позиция, свидетельствующая о том, что агент свободен, приобретает фишку со значением *done* (задача выполнена), переход захвата агента активизируется, перемещая следующую фишку на позицию использования агента. Процесс обработки задач прекращается в случае, если не остается фишек в позиции *wait* [3].

При моделировании сложных систем на базе аппарата сетей Петри задается определенное количество входных задач и анализируется поведение системы. На основе выходных данных формируются статистические результаты.

При анализе сложных систем с использованием аппарата сетей Петри стоит выделить такие основные свойства, как ограниченность, безопасность, сохраняемость, достижимость, живость [4].

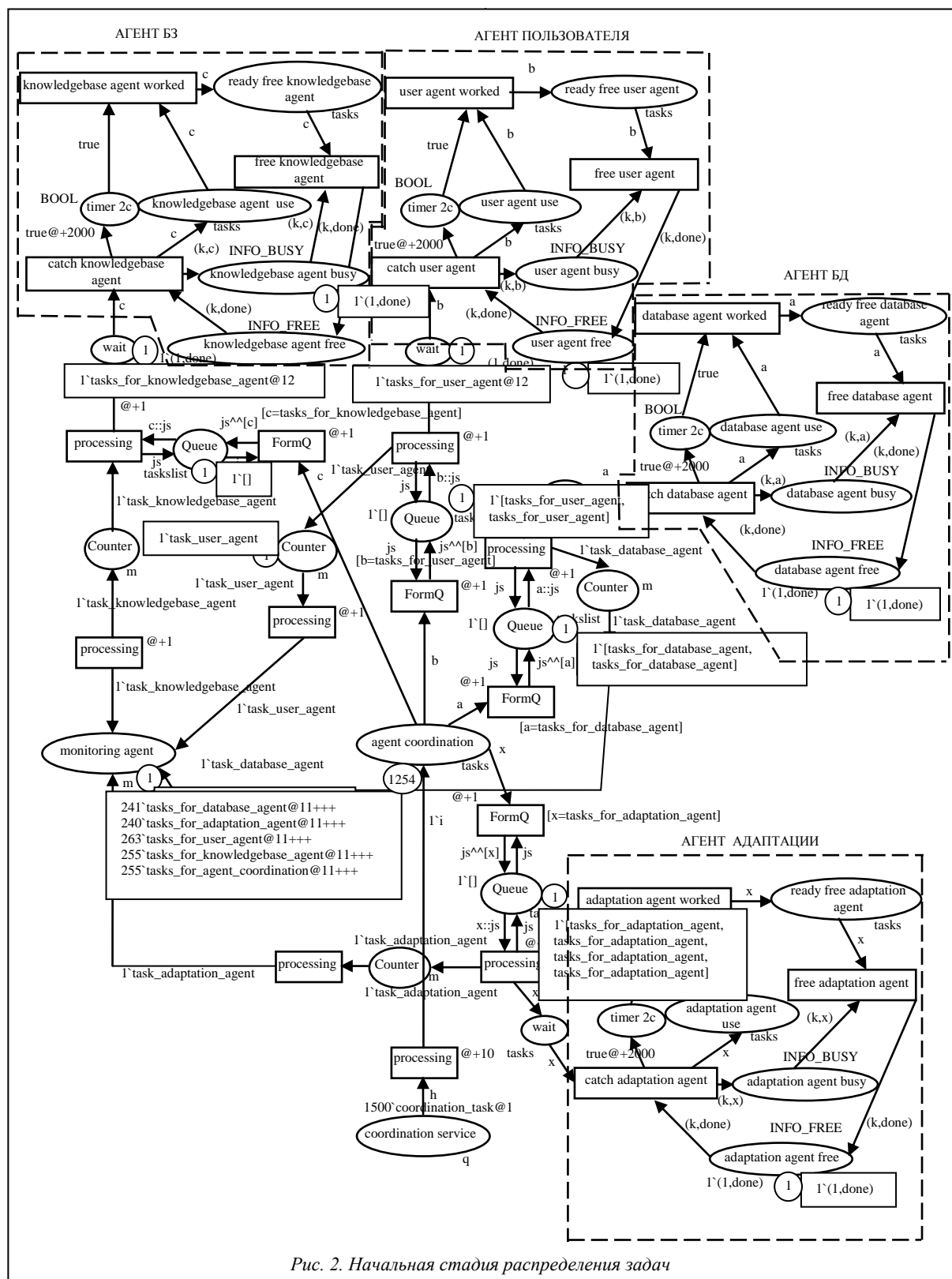


Рис. 2. Начальная стадия распределения задач

Анализируя свойства вышеупомянутой модели, можно сделать вывод, что она является

- неограниченной, так как число меток в любой позиции рассматриваемой сети может превысить некоторое максимальное значение K ;
- небезопасной, так как $K \neq 1$;

- несохраняемой из-за непостоянства загрузки ресурсов; число входов в каждый переход не равняется числу выходов;

- достижимой в силу наличия возможности перехода сети из одного заданного состояния (характеризуемого распределением меток) в другое;

- живой, так как любой из переходов срабатывает при поступлении задач.

Наличие данных свойств у системы обусловлено заданной архитектурой и большими объемами поставленных задач.

Опишем агента как кортеж: $A = \langle P_I, P_O, P_{INT}, T_{INT}, F, M_0 \rangle$, где P_I – множество входных портов агента; P_O – множество выходных мест; P_{INT} – множество внутренних мест агента; T_{INT} – множество внутренних переходов агента; F – отношение

инцидентности; M_0 – начальная разметка внутренних мест агента. Кортеж $\langle P_I, UP_O, UP_{INT}, T_{INT}, F, M_0 \rangle$ является обыкновенной сетью Петри.

Рассмотрим механизм получения оценки качества построения модели системы с использованием элементов нечеткой логики. В процессе построения нечеткой модели были выделены 34 продукционных правила. Для построения модели оценки качества выделены следующие нечеткие лингвистические переменные: *поставленные за-*

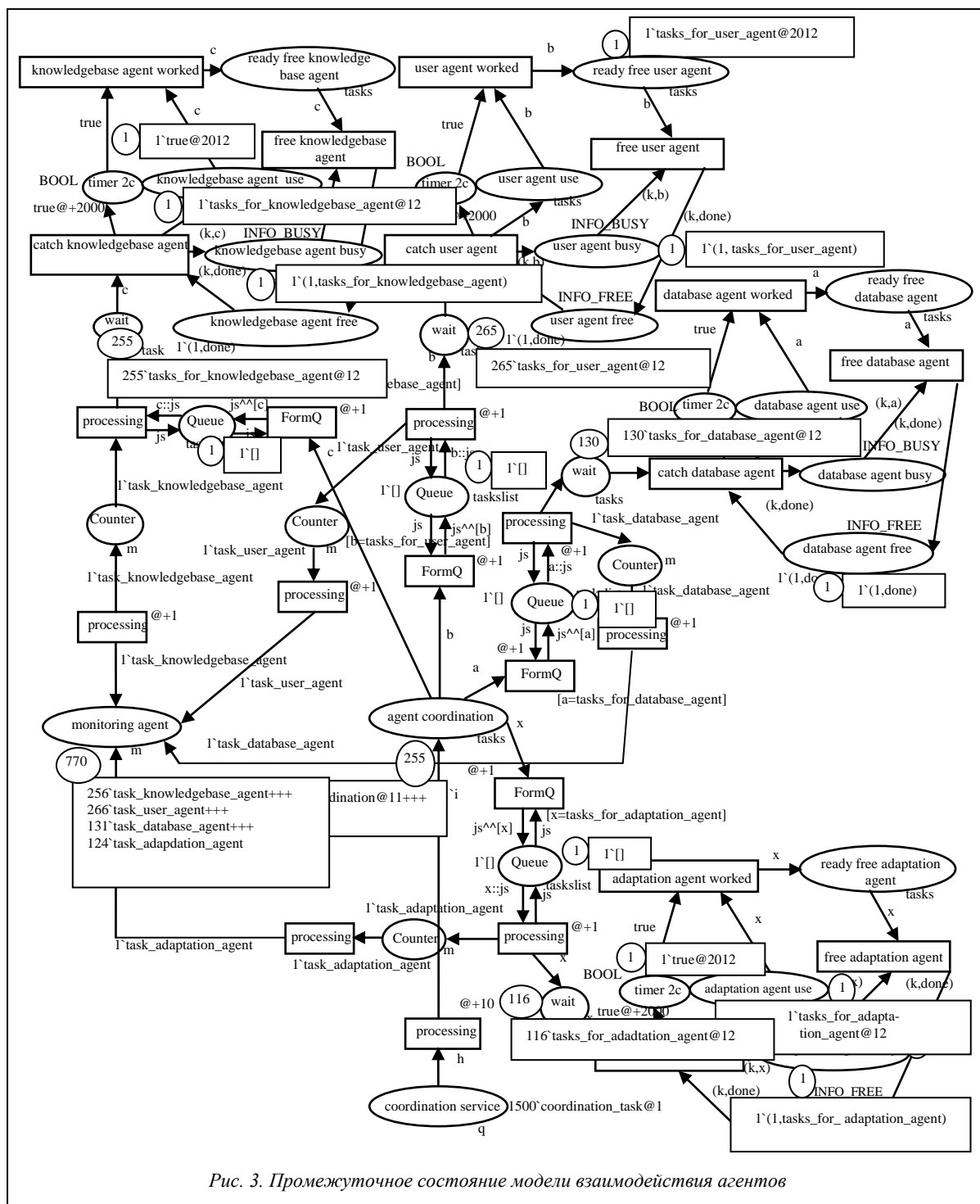


Рис. 3. Промежуточное состояние модели взаимодействия агентов

дачи (*tasks_in_view*), выполненные задачи (*done_tasks*), время выполнения (*time*), качество построения (*build_quality*). Построение модели оценки качества выполним в среде FuzzyTech (рис. 4).

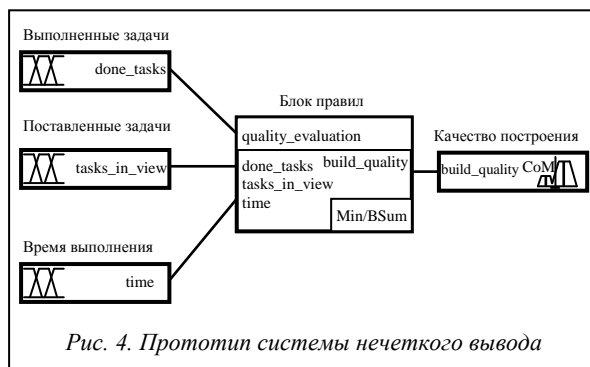


Рис. 4. Прототип системы нечеткого вывода

В качестве терм-множества лингвистической переменной *поставленные задачи* используется множество $T_1 = ["zero"(\text{нуль}), "little"(\text{мало}), "medium"(\text{средне}), "many"(\text{много})]$, в качестве терм-множества лингвистической переменной *выполненные задачи* – множество $T_2 = ["zero"(\text{нуль}), "little"(\text{мало}), "medium"(\text{средне}), "many"(\text{много})]$, в качестве терм-множества лингвистической переменной *время выполнения* – множество $T_3 = ["zero"(\text{нуль}), "quickly"(\text{очень быстро}), "quick"(\text{быстро}), "long"(\text{долго}), "verylong"(\text{очень долго})]$, в

качестве терм-множества выходной лингвистической переменной *качество построения* – множество $T_4 = ["v_low_quality"(\text{очень низкое качество}), "low_quality"(\text{низкое качество}), "zero"(\text{нуль}), "high_quality"(\text{высокое качество}), "v_high_quality"(\text{очень высокое качество})]$. Оценка качества выставляется по шкале от -10 до 10, при которой цифре -10 будет соответствовать наихудшая оценка качества, а цифре 10 – наилучшая оценка (рис. 5).

Используя ранее полученные данные о среднем количестве выполняемых системой задач, получаем результат, свидетельствующий о том, что качество построения оценивается как высокое ($build_quality=5$ при $time=30$, $done_tasks=600$, $tasks_in_view=1000$).

Отсутствие областей значений входных лингвистических переменных, в которых малые изменения значений входных переменных модели приводят к большим (скачкообразным) изменениям значений выходной лингвистической переменной, а также отсутствие разрывов на поверхности нечеткого вывода свидетельствуют о достаточном уровне устойчивости разработанной нечеткой модели (рис. 6) [5].

Полученные результаты моделирования могут быть применены при построении агентной составляющей адаптивной информационной системы, оценке качества функционирования на предпри-

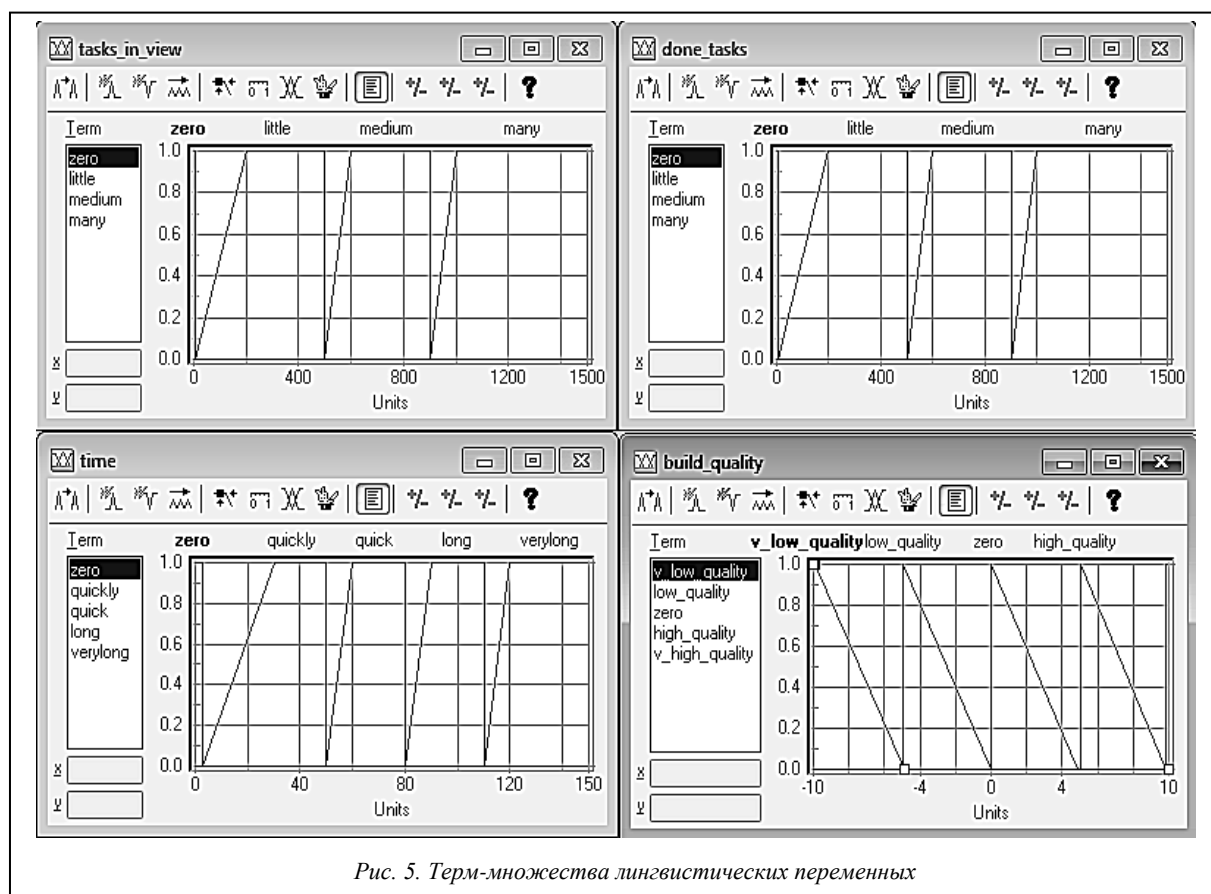


Рис. 5. Терм-множества лингвистических переменных

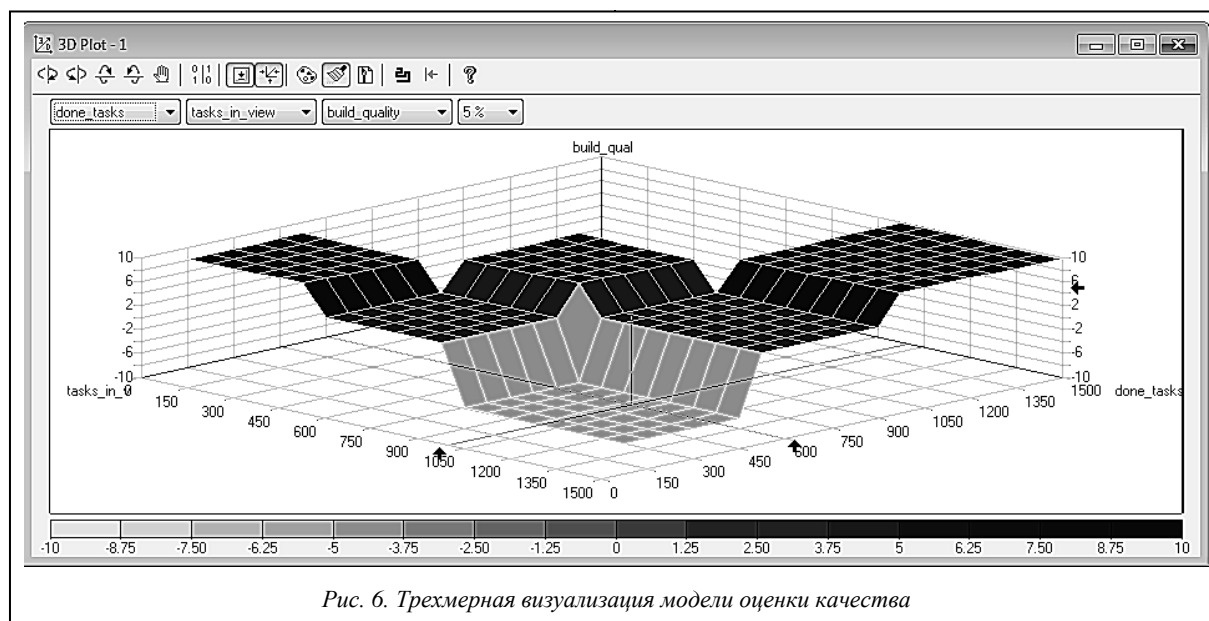


Рис. 6. Трехмерная визуализация модели оценки качества

ятиях или в их отделах, оперирующих значительными объемами информации.

Сети Петри позволяют отражать динамические характеристики моделей. Аппарат цветных сетей Петри, реализуемый в среде CPN Tools, позволяет смоделировать и проанализировать работу системы с учетом ее архитектуры, выявить наиболее загруженные элементы системы.

Литература

1. Владимиров А.В. Общая модель взаимодействия интеллектуальных агентов в многоагентной адаптивной информационной системе. Вузовская наука – региону: матер. 10-й Всерос. науч.-технич. конф. Вологда: ВоГТУ, 2012.
2. Зайцев Д.А., Шмелева Т.Р. Моделирование телекоммуникационных систем в CPN Tools: учеб. пособие. Одесса: ОНАС, 2008. 68 с.
3. Шахов В.Ю. Моделирование программно-аппаратных «реактивных» систем раскрашенными сетями Петри. Долгопрудный, 2005. URL: <http://www.softcraft.ru/theory/mrs/index.shtml> (дата обращения: 18.03.2013).
4. Кириличев Б.В. Моделирование систем: учеб. пособие. М.: МГИУ, 2009. 274 с.

5. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб: БХВ-Петербург, 2005. 736 с.

References

1. Vladimirov A.V. A general model of intelligent agents' interaction in multi-agent adaptive information system. *Vuzovskaya nauka – regionu: Materialy 10 Vseross. nauchno-tekhnich. konf.* [University science to regions: proc. of the 10 All-Russian science and technical conf.]. Vologda, Vologda State Tech. Univ. Publ., 2012.
2. Zaytsev D.A., Shmeleva T.R. *Modelirovanie telekommunikatsionnykh sistem v CPN Tools* [Modeling telecommunication systems with CPN Tools]. Study guide, Odessa, ONAS Publ., 2008, 68 p.
3. Shakhov V.Yu. *Modelirovanie programmno-apparatnykh "reaktivnykh" sistem raskrashennymi setyami Petri* [Modeling hardware and software "reactive" systems with coloured Petri nets]. Dolgoprudny, 2005. Available at: <http://www.softcraft.ru/theory/mrs/index.shtml> (accessed 18 March 2013).
4. Kirilichev B.V. *Modelirovanie sistem: ucheb. posobie* [Modeling systems]. Study guide, Moscow, Moscow State Industrial Univ. Publ., 2009, 274 p. (in Russ.).
5. Leonenkov A.V. *Nechetkoe modelirovanie v srede MATLAB i fuzzyTECH* [Fuzzy modeling in MATLAB and fuzzyTECH]. St. Petersburg, BHV-Peterburg Publ., 2005, 736 p.