

Розглянуто задачу розробки моделі і методу вибору варіанту опису раціональної архітектури інформаційної системи. Розроблені моделі і метод дозволяють Постачальнику та Споживачу ІТ-послуг обирати варіант опису архітектури системи, який найбільш вдовольняє висунуті цими сторонами функціональні вимоги до створюваної системи та враховує повторне використання рішень, що реалізують висунуті вимоги

Ключові слова: інформаційна система, функціональні вимоги, алгоритм CLOPE, функція цілі, біматрична гра, рівновага за Нешем

Рассмотрена задача разработки модели и метода выбора варианта описания рациональной архитектуры информационной системы. Разработанная модель и метод позволяют Поставщику и Потребителю ИТ-услуг выбирать вариант описания архитектуры системы, лучше всего удовлетворяющий выдвинутые этими сторонами функциональные требования к создаваемой системе. Кроме того, выбранный вариант позволяет учесть повторное использование решений, реализующих выдвинутые требования

Ключевые слова: информационная система, функциональные требования, алгоритм CLOPE, функция цели, биматричная игра, равновесие по Нэшу

УДК 044.03; 658.11.05.06

DOI: 10.15587/1729-4061.2016.60583

РАЗРАБОТКА МОДЕЛИ И МЕТОДА ВЫБОРА ОПИСАНИЯ РАЦИОНАЛЬНОЙ АРХИТЕКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ

М. В. Евланов

Кандидат технических наук, доцент

Кафедра информационных

управляющих систем

Харьковский национальный

университет радиозлектроники

пр. Ленина, 14, г. Харьков, Украина, 61166

E-mail: iyc@kture.kharkov.ua

1. Введение

Современное представление жизненного цикла (ЖЦ) информационной системы (ИС) определяется стандартом ISO/IEC 15288 «System Engineering – System life cycle processes» на основе возможного использования абстрактной функциональной модели. Такая модель определяет концептуализацию потребности в системе, ее реализации, применения, развития и ликвидации [1]. При построении такой модели предполагается, что система развивается на протяжении ЖЦ в результате неких действий. Эти действия осуществляются и управляются людьми, которые работают в организациях и используют определенные процессы в своей деятельности. Эти процессы называются процессами сами ЖЦ. Детали модели ЖЦ выражаются в терминах процессов ЖЦ, их результатов, взаимосвязи и возникновения.

На ранних стадиях ЖЦ системы выполняются процессы определения требований правообладателей, анализа требований и проектирования архитектуры системы [1]. Взаимодействие данных процессов в ходе макропроектирования ИС показано в виде IDEF3-модели на рис. 1.

Исследование описаний процессов ЖЦ ИС, показанных на рис. 1, позволяет утверждать, что системное описание архитектуры создаваемой системы является первым системным описанием, которое должно фор-

мировать представление о создаваемой ИС в целом. Данное описание в [1] рекомендуется представлять в виде итогового проекта архитектуры системы, формируемого на основе множества приемлемых архитектур. Описание архитектуры создаваемой системы формируется в ходе выполнения процессов ЖЦ ИС на основе описаний множества отдельных требований правообладателей и системных требований. Системное описание архитектуры связывает описания отдельных требований к системе с решениями по отдельным элементам системы и решениями по обеспечивающим системам [1, 2].

Множество требований правообладателей является хронологически первым системным описанием создаваемой ИС. В то же время данное множество представляет систему не как единый целостный продукт, а как набор не связанных друг с другом описаний отдельных элементов системы. Данное множество включает в себя самые различные требования, на основе которых формируется множество сценариев, описывающих отдельную функциональную возможность создаваемой системы. Требования правообладателей и сценарии, идентифицирующие функциональные возможности системы, выражаются в терминах моделей (текстовых или формальных). Подобные модели ориентированы на цели и поведение системы и описывающих систему в контексте среды и условий функционирования.

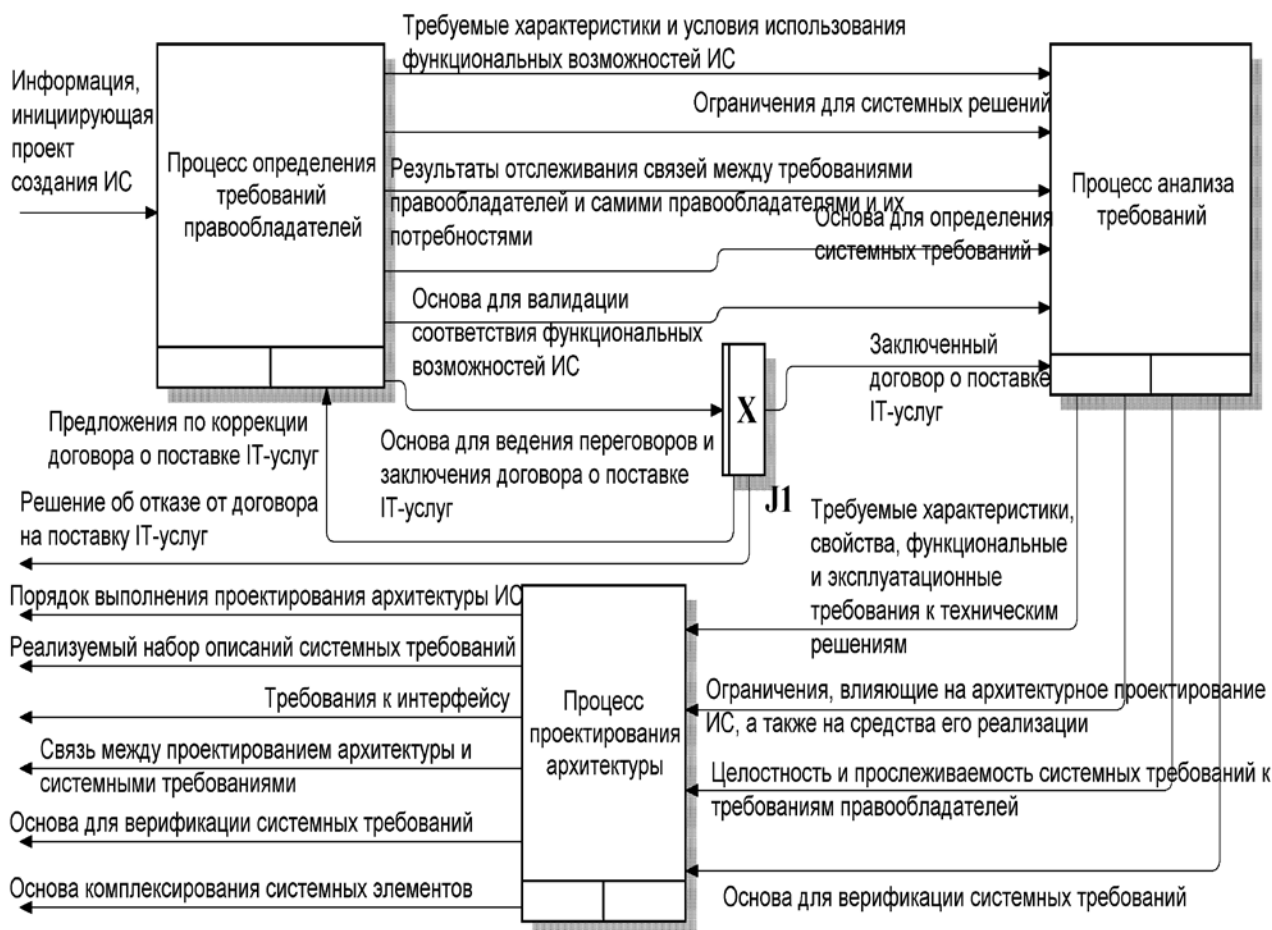


Рис. 1. IDEF3-модель взаимосвязи процессов информационной системы, выполняемых на ранних стадиях ее жизненного цикла

Сказанное выше позволяет сделать следующие выводы:

а) рекомендуемые стандартами [1, 2] системные описания, формируемые на ранних стадиях ЖЦ ИС, ориентированы на представление создаваемой системы как множества отдельных требований, функций и архитектурных элементов, а не единого целостного проекта;

б) в ходе выполнения процессов формирования требований правообладателей, анализа требований и проектирования архитектуры задачи системной интеграции не решаются и не ставятся;

в) использование в ходе формирования системных описаний опыта, накопленного в ходе выполнения предыдущих проектов создания аналогичных систем, не рассматривается;

г) решение о целесообразности начала IT-проекта создания ИС должно приниматься на основе множества отдельных требований правообладателей;

д) оценки объема работ по созданию ИС сильно искажены вследствие невозможности учета опыта создания аналогичных ИС и отказа от представления ИС как единого целостного продукта.

Данные выводы позволяют утверждать о существовании серьезных проблем в ходе создания ИС на ранних стадиях ее ЖЦ. Данные проблемы значительно повышают затраты на создание ИС и увеличивают

количество инцидентов, которые могут возникнуть в ходе разработки обеспечивающей части данной ИС. В то же время практика выполнения IT-проектов показывает [3, 4], что изменение ИС под влиянием заинтересованных сторон наиболее эффективно и требует наименьших затрат преимущественно на ранних стадиях ее ЖЦ. Поэтому решение указанных проблем является одним из наиболее перспективных направлений научно-прикладных исследований в IT-сфере.

2. Анализ литературных данных и постановка проблемы

В настоящее время основное внимание исследователей в области описаний требований к ИС сосредоточено на разработке методов, позволяющих создавать и обрабатывать формальные модели требований к системе. Под моделью требования понимается комплекс описаний требований теми способами, которые позволяют на основе данных описаний выполнить необходимые работы в рамках ЖЦ ИС с применением конкретных методологий и информационных технологий [5]. В качестве примеров таких исследований можно указать:

а) описание применения разработанного метода для анализа требований к ERP-ИС, выполненное в [6];

б) вопросы создания и применения метода для сценарно-основанного анализа требований к системе, рассмотренные в [7];

в) применение процессной модели для выявления требований к специализированной ИС выявления и обработки знаний из бизнес-информации, описанное в [8];

г) вариант решения проблемы автоматизации работ по формализации требований к ИС, рассмотренный в [9].

Отдельным перспективным направлением в области формирования и анализа требований к ИС является исследование и разработка моделей и методов добычи знаний из описаний потребностей и требований к системе. Один из вариантов формализации предметной области (ПрО) корпоративных ИС путем семантического моделирования рассматривается в [10]. В [11] рассматривается вопрос применения методов добычи знаний для преобразования высказываемых потребностей в множество описаний требований к ИС в здравоохранении.

Однако анализ подобных исследований показывает, что проблема преобразования неформальных потребностей в формальные модели требований решается в них преимущественно на концептуальном уровне [12]. Одной из главных причин этого следует считать отсутствие единого определения понятия «требование к системе» [13].

В то же время существует достаточно большой практический опыт разработки и эксплуатации ИТ-продуктов, используемых для формирования, анализа и управления требованиями к системе – систем управления требованиями. Наиболее характерными примерами таких систем являются продукты компании IBM – ставший классическим (2004 г.) IBM Rational RequisitePro [14] и современный продукт (2013–2014 гг.) IBM Rational DOORS Next Generation [15]. Среди систем управления требованиями других разработчиков следует отметить такие продукты, как 3SL Cradle [16] и Devprom Requirements [17].

Анализ этого опыта в [18] показывает существующее разделение систем управления требованиями на два различных направления. Продукты первого направления работают с требованиями априорно заданных типов и не могут быть расширены или адаптированы под требования конкретных пользователей. Продукты второго направления основаны на принципе открытых моделей и позволяют описывать не только требования к создаваемой системе, но и другие артефакты, используемые для описания системы (например, риски, тесты, ошибки и т.п.). В общем случае продукты второго направления ориентированы на хранение следующих видов описания создаваемой системы:

- а) исходные требования (потребности);
- б) производные требования – текстовые или графические спецификации;
- в) модели реализации системы и ее элементов.

Однако ни одна из существующих систем управления требованиями не ориентирована на автоматизацию синтеза описания архитектуры создаваемой системы. Кроме того, в данных системах остается практически нерешенной проблема принятия решений о повторном использовании ранее реализованных требований в новых ИТ-проектах [18].

В целом по результатам проведенного анализа можно сделать следующий вывод: к настоящему времени проблема формального описания требований, методов их формирования и анализа – в том числе и на уровне добываемых из требований знаний – не имеет приемлемого решения. Те же формальные представления требований, которые используются для разработки систем управления требованиями, ориентированы на описание отдельных требований как артефактов создаваемой или модернизируемой ИС. Поэтому попытки синтеза ИС на основе решений, реализующих отдельные требования, чаще всего порождают эффект «ИТ-слепоты» – неспособность существующих ИС и ИТ «увидеть» и оценить реальные процессы в той среде, в которую они включены [19].

3. Цель и задачи исследования

Целью работы является усовершенствование способов выполнения процесса проектирования архитектуры создаваемой ИС, позволяющих автоматизировать решение совокупности задач синтеза описания архитектуры системы.

Наиболее значимой нерешенной частью данной проблемы является задача формирования вариантов описаний архитектуры создаваемой ИС и выбора из них варианта, который в максимальной степени удовлетворит бы заинтересованные стороны ИТ-проекта создания ИС. Заинтересованными сторонами для ИТ-проекта создания ИС являются Поставщик и Потребитель. Данная задача должна решаться на основе формальных описания отдельных требований к ИС. Поэтому в качестве подхода к решению данной задачи предлагается использовать рассмотренный в [4] сервисный подход к созданию ИС и созданные на его основе модели паттернов проектирования требований к ИС. Эти модели позволяют устанавливать взаимно-согласованную структуру формальных описаний (представлений) требований на уровнях данных, информации и знаний.

Для достижения поставленной цели в статье рассматривается решение следующих задач:

- выбор метода синтеза вариантов описания архитектуры создаваемой ИС на основе формальных представлений требований к ИС на уровне знаний;
- формальное описание целей заинтересованных сторон ИТ-проекта создания ИС в ходе проектирования архитектуры системы;
- разработка формального описания модели выбора варианта описания рациональной архитектуры создаваемой ИС;
- разработка описания метода выбора варианта описания рациональной архитектуры создаваемой ИС.

4. Выбор метода синтеза вариантов описания архитектуры информационной системы

Для решения задачи выбора варианта описания рациональной архитектуры создаваемой ИС предлагается в качестве исходных данных использовать следующие модели, предложенные автором в [20]:

- а) модель представления i -го функционального требования к ИС на уровне знаний Потребителя K_i^{fu} ,

формируемая в результате выполнения соответствующего метода;

б) модель представления i -го функционального требования к ИС на уровне знаний Поставщика K_i^{fr} , формируемая в результате выполнения соответствующего метода;

в) модель общесистемного представления i -го функционального требования к ИС на уровне знаний K_i^{fs} , формируемая в результате выполнения соответствующего метода.

Эти модели являются частными случаями формального описания представления i -го функционального требования к ИС на уровне знаний tr_i^f [20], которое определяется соответствующими паттернами [21] и представляет собой множество кортежей следующего вида:

$$K_i^f = \{ \langle d_n^{ij}, \{ \langle d_{el_fr_t}^{ij}, d_{el_fr_t}^{ij} \rangle \}, \langle d_g^{ij}, \{ \langle d_{el_if}^{ij}, d_{el_if_t}^{ij} \rangle \}, \langle d_{fr_rel_n}^{ij}, \{ \langle d_{el_fr_rel}^{ij}, d_{el_fr_rel_t}^{ij} \rangle \} \rangle \}, \quad (1)$$

где d_n^{ij} – описание наименования фрейма; $d_{el_fr}^{ij}$ – описание элемента фрейма; $d_{el_fr_t}^{ij}$ – описание типа элемента фрейма; d_g^{ij} – описание наименования интерфейса; $d_{el_if}^{ij}$ – описание элемента интерфейса; $d_{el_if_t}^{ij}$ – описание типа элемента интерфейса; $d_{fr_rel_n}^{ij}$ – описание наименования связи между интерфейсами и/или фреймами; $d_{el_fr_rel}^{ij}$ – описание элемента связи; $d_{el_fr_rel_t}^{ij}$ – описание типа элемента связи.

Исходным вариантом описания архитектуры будем называть сеть фреймов [21] $Arch_{base}$, формируемую следующим образом:

$$Arch_{base} = \bigcup_{i=c+1}^e K_i^{fs}, \quad (2)$$

где $c+1$ – номер первого требования из множества функциональных требований к ИС; e – номер последнего требования из множества функциональных требований к ИС.

Сеть фреймов $Arch_{base}$ описывает вариант архитектуры создаваемой ИС, в котором каждое функциональное требование будет описывать отдельную ИТ-услугу (для данного варианта считается, что все функциональные требования сформулированы правильно).

Для выявления других возможных вариантов описания архитектуры ИТ-услуг создаваемой ИС следует учитывать такие особенности формируемых сетей фреймов:

а) в соответствии с рассмотренным в [21] множеством паттернов описания формируемых сетей фреймов предполагается сохранять в специализированном хранилище данных как совокупность транзакций, каждая из которых описывает отдельное представление K_i^{fs} ;

б) представление K_i^{fs} является категорийными данными (формализованные структурированные описания фреймов, интерфейсов и связей), числовые представления которых затруднено;

в) оценка степени дублирования представлений K_i^{fs} должна основываться на оценке количества повторений описаний элементов сети фреймов в различных представлениях;

г) выявление дублирования функциональных требований должно приводить к синтезу новых вариантов описаний архитектуры, в которых описания новых ИТ-услуг формируются путем группировки исходных представлений K_i^{fs} .

Исходя из отмеченных особенностей, задачу синтеза вариантов архитектуры создаваемой ИС следует рассматривать как задачу кластеризации категориальных данных, алгоритм решения которой основан на оптимизации некоего глобального критерия. В качестве такого алгоритма предлагается использовать алгоритм CLOPE [22]. Применение этого алгоритма позволяет формализовать задачу синтеза вариантов описаний архитектуры создаваемой ИС следующим образом.

Пусть имеется сеть фреймов (2), состоящая из множества представлений K_i^{fs} . Каждое представление K_i^{fs} является набором объектов ob (фреймов, интерфейсов и связей), описание которого аналогично (1). Множество искомым описаний ИТ-услуг IT_{acm} есть разбиение сети фреймов (2), такое, что $\{IT_{acm_1}, \dots, IT_{acm_k}\} = Arch_{base}$ и $IT_{acm_i} \neq \emptyset \wedge IT_{acm_i} \cap IT_{acm_j} = \emptyset$ для $1 \leq i, j \leq k$, где k – количество ИТ-услуг создаваемой ИС. Каждая ИТ-услуга IT_{acm_j} имеет следующие характеристики:

- $D(IT_{acm_j})$ – множество уникальных объектов;
- $Occ(ob, IT_{acm_j})$ – количество вхождений объекта ob в описание ИТ-услуги IT_{acm_j} ;
- $S(IT_{acm_j}) = \sum_{ob \in D(IT_{acm_j})} Occ(ob, IT_{acm_j})$;
- $W(IT_{acm_j}) = |D(IT_{acm_j})|$;
- $H(IT_{acm_j}) = S(IT_{acm_j}) / W(IT_{acm_j})$.

Тогда расчет функции стоимости ИТ-услуги $Profit(IT_{acm}, r)$ в соответствии с положениями алгоритма CLOPE будет проводиться по формуле [22]:

$$Profit(IT_{acm}, r) = \frac{\sum_{j=1}^k \frac{S(IT_{acm_j})}{W(IT_{acm_j})^r} \times |IT_{acm_j}|}{\sum_{j=1}^k |IT_{acm_j}|}, \quad (3)$$

где $|IT_{acm_j}|$ – количество общесистемных представлений функциональных требований на уровне знаний K_i^{fs} , описывающих ИТ-услугу IT_{acm_j} .

Тогда постановку задачи синтеза вариантов описаний архитектуры, создаваемой ИС, можно сформулировать следующим образом: для заданных $D(Arch_{base})$ и r найти разбиение: $IT_{acm} : Profit(IT_{acm}, r) \rightarrow \max$.

Однако не следует забывать, что множество сформулированных требований практически никогда не описывает создаваемую ИС полностью [22]. Поэтому приведенную в предыдущем абзаце постановку задачи следует модифицировать таким образом: для заданных $D(Arch_{base})$ и r найти разбиения:

$$IT_{acm} : Profit(IT_{acm}, r) \in [Profit_{\max} - \epsilon; Profit_{\max}],$$

здесь $Profit_{\max}$ – максимальное значение функции (3), ϵ – величина допустимой погрешности.

Если значения величины ϵ нельзя определить с помощью методов экспертного оценивания, то реко-

мендуется рассчитать ее по следующему правилу: $\epsilon = 0,1 \times \text{Profit}_{\max}$.

Точное определение значения коэффициента отталкивания γ требует проведения специальных исследований успешных IT-проектов создания ИС. Основываясь на результатах исследований законов композиции функциональной структуры ИС, рассмотренных в [23], рекомендуется выбирать значение γ , исходя из приведенных в табл. 1 данных.

Таблица 1

Зависимость значения коэффициента отталкивания γ от вариантов организации архитектуры информационной системы, создаваемой как совокупность IT-услуг

Вариант организации архитектуры информационной системы	Значение коэффициента отталкивания γ
Функциональная (подсистемы)	1,5
Модульная	2
Сервисная	≥ 3

Результатом решения поставленной выше задачи является множество приемлемых вариантов описаний архитектуры ИС, в которых степень дублирования описаний ПрО в функциональных требованиях сведена до необходимого минимума. Каждый из этих вариантов описаний представляет собой множество IT-услуг IT_{acm} .

В свою очередь, каждая из IT-услуг IT_{acm_j} является результатом объединения, аналогичного (2) отдельных представлений K_i^{fs} , включаемых в описание данной IT-услуги. Данный метод позволяет выделять для последующего анализа варианты описания архитектуры создаваемой ИС, которые лишь незначительно хуже оптимального, что позволяет значительно расширить область поиска компромиссного варианта архитектуры создаваемой ИС, удовлетворяющего как Поставщика, так и Потребителя.

5. Результаты разработки формального описания целей заинтересованных сторон IT-проекта

Выполнение процесса проектирования архитектуры системы предлагается осуществлять, исходя из сформулированных в [21, 24] описаний глобальных целей Поставщика и Потребителя. Так, глобальная цель деятельности Поставщика в процессе проектирования архитектуры определяется следующим образом: «Целью Поставщика является предоставление Потребителю такого набора IT-услуг, который наилучшим образом соответствует комплексу требований, определенных Потребителем и особенностями бизнес-процессов Потребителя, при ограничениях на стоимость, время выполнения, качество и содержание работ по предоставлению данного набора IT-услуг Потребителю». Глобальная цель деятельности Потребителя в процессе проектирования архитектуры определяется следующим образом: «Целью Потребителя является поиск и организация взаимодействия с таким Поставщиком, который предоставляет набор IT-услуг, наилучшим образом соответствующий комплексу требований, определенных Потребителем и особенностями биз-

нес-процессов Потребителя, при ограничениях на стоимость, время выполнения, качество и содержание работ по предоставлению данного набора IT-услуг, выбранных Поставщиком».

Степень удовлетворения Поставщика в ходе выполнения процесса проектирования архитектуры ИС на основе выдвинутых к системе функциональных требований может быть определена следующим образом:

$$r^{\text{Pr}}(K_i^{\text{fr}}) = f(K_i^{\text{fs}}, K_i^{\text{fr}}) = 1 - \frac{|K_i^{\text{fs}} / K_i^{\text{fr}}|}{|K_i^{\text{fs}}|} \tag{4}$$

Степень удовлетворения Потребителя в ходе выполнения процесса проектирования архитектуры ИС на основе выдвинутых к системе функциональных требований может быть определена следующим образом:

$$r^{\text{U}}(K_i^{\text{fu}}) = f(K_i^{\text{fs}}, K_i^{\text{fu}}) = 1 - \frac{|K_i^{\text{fs}} / K_i^{\text{fu}}|}{|K_i^{\text{fs}}|} \tag{5}$$

Следует учесть, что общесистемное представление i -го функционального требования к ИС на уровне знаний K_i^{fs} в случае выявления дублирования отдельных функциональных требований $s+1 < m, n < e, m \neq n$ формируется следующим образом:

$$K_i^{\text{fs}} = K_{(m \cup n)}^{\text{fs}} = K_m^{\text{fs}} \cup K_n^{\text{fs}} \tag{6}$$

Тогда глобальные цели Поставщика и Потребителя можно описать следующими функциями:

$$F_{\text{Pr}} = \sum_{i=c+1}^e 1 - \frac{|K_i^{\text{fs}} / K_i^{\text{fr}}|}{|K_i^{\text{fs}}|} \rightarrow \max, \tag{7}$$

$$F_{\text{U}} = \sum_{i=c+1}^e 1 - \frac{|K_i^{\text{fs}} / K_i^{\text{fu}}|}{|K_i^{\text{fs}}|} \rightarrow \max. \tag{8}$$

Физический смысл функции (7) заключается в максимизации повторного использования в IT-проекте создания ИС IT-сервисов, реализующих сформулированные функциональные требования к этой системе. В этом случае основным видом IT-проектов создания ИС с точки зрения Поставщика будет адаптация типовых ИС, IT-услуг и реализующих эти услуги IT-сервисов к особенностям процессов Потребителя. Создание ИС «с нуля» будет рассматриваться Поставщиком как частный, наименее желаемый случай IT-проекта, который, однако, после успешного завершения может существенно расширить возможности Поставщика.

С точки зрения Потребителя основной эффект от выполнения IT-проекта создания требуемой ИС и, следовательно, основной выигрыш, описываемый функцией (8), достигается в случае максимизации выполнения требований к ИС, выдвинутых Потребителем и принятых к реализации Поставщиком. При этом Потребитель интересуется в первую очередь реализация описаний онтологий именно тех терминов ПрО, которые характерны для его процессов.

6. Результаты разработки модели и метода выбора варианта описания рациональной архитектуры создаваемой информационной системы

Достижение цели (7) Поставщиком возможно в результате выполнения следующих действий [21]:

- разработка новых ИТ-услуг и ИТ-сервисов, удовлетворяющих требования, выдвигаемые Потребителями;
- адаптация разработанных ранее ИТ-услуг и ИТ-сервисов под особенности требований конкретных Потребителей.

Достижение цели (8) Потребителем возможно в результате выполнения следующих действий [21]:

- формулировка требований к ИС, ИТ-услугам и ИТ-сервисам, учитывающим общие и индивидуальные особенности бизнес-процессов объекта автоматизации;

- анализ и выбор ИТ-услуг и ИТ-сервисов, предлагаемых Поставщиками и соответствующих сформулированным требованиям.

При этом между Поставщиком и Потребителем в процессе проектирования архитектуры ИС в общем случае практически отсутствуют какие-либо формы предварительной договоренности, ограничивающие выбор Потребителя или же разработки Поставщика. Это позволяет представить процесс проектирования архитектуры ИС как рационального набора ИТ-услуг в виде игры Поставщика и Потребителя [21]. Эта игра будет иметь следующие отличительные особенности:

- по количеству игроков – игра двух лиц (Поставщика и Потребителя ИТ-услуг);
- по количеству стратегий – конечная игра, число чистых стратегий которой ограничено числом отдельных ИТ-услуг;
- по типу взаимоотношений игроков – бескоалиционная игра, до окончания которой Поставщик и Потребитель не могут действовать совместно (вступать в соглашения, передавать информацию, ресурсы и пр.);
- по характеру выигрышей – игра с нулевой суммой (сумма выигрышей Поставщика и Потребителя в каждой партии игры равняется нулю; общий их капитал перераспределяется между ними независимо от полученных результатов);
- по виду функции выигрыша – биматричная игра, поскольку функции выигрыша Поставщика и Потребителя различны и определяются, как следует из (6) и (7), различными функциями выигрыша.

В общем случае такая игра для Поставщика и Потребителя будет иметь вид [21]:

$$\Gamma_{IS} = \langle \{Pr, U\}, \{X^{(Pr, U)}\}, \{f^{(Pr, U)}\} \rangle, \quad (9)$$

где Γ_{IS} – обозначение игры Поставщика и Потребителя; $\{Pr, U\}$ – множество игроков, участвующих в игре Γ_{IS} (Поставщик и Потребитель); $\{X^{(Pr, U)}\}$ – множество стратегий игры Γ_{IS} , формируемое на основе множества вариантов описаний архитектуры создаваемой ИС

$\Gamma_{acm}; \{f^{(Pr, U)}\}$ – множество функций выигрыша игры Γ_{IS} , состоящее из выражений (7) и (8).

Матрица выигрыша Поставщика будет иметь следующий вид:

$$Pr = \begin{pmatrix} 1 - \frac{|K_{I(c+1)}^{f_{IS}} / K_{I(c+1)}^{f_{Pr}}|}{|K_{I(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Ii}^{f_{IS}} / K_{Ii}^{f_{Pr}}|}{|K_{Ii}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Ie}^{f_{IS}} / K_{Ie}^{f_{Pr}}|}{|K_{Ie}^{f_{IS}}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{J(c+1)}^{f_{IS}} / K_{J(c+1)}^{f_{Pr}}|}{|K_{J(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Ji}^{f_{IS}} / K_{Ji}^{f_{Pr}}|}{|K_{Ji}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Je}^{f_{IS}} / K_{Je}^{f_{Pr}}|}{|K_{Je}^{f_{IS}}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{K(c+1)}^{f_{IS}} / K_{K(c+1)}^{f_{Pr}}|}{|K_{K(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Ki}^{f_{IS}} / K_{Ki}^{f_{Pr}}|}{|K_{Ki}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Ke}^{f_{IS}} / K_{Ke}^{f_{Pr}}|}{|K_{Ke}^{f_{IS}}|} \end{pmatrix}. \quad (10)$$

Матрица выигрыша Потребителя будет иметь следующий вид:

$$U = \begin{pmatrix} 1 - \frac{|K_{I(c+1)}^{f_{IS}} / K_{I(c+1)}^{f_{U}}|}{|K_{I(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Ii}^{f_{IS}} / K_{Ii}^{f_{U}}|}{|K_{Ii}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Ie}^{f_{IS}} / K_{Ie}^{f_{U}}|}{|K_{Ie}^{f_{IS}}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{J(c+1)}^{f_{IS}} / K_{J(c+1)}^{f_{U}}|}{|K_{J(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Ji}^{f_{IS}} / K_{Ji}^{f_{U}}|}{|K_{Ji}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Je}^{f_{IS}} / K_{Je}^{f_{U}}|}{|K_{Je}^{f_{IS}}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{K(c+1)}^{f_{IS}} / K_{K(c+1)}^{f_{U}}|}{|K_{K(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Ki}^{f_{IS}} / K_{Ki}^{f_{U}}|}{|K_{Ki}^{f_{IS}}|} & \dots & 1 - \frac{|K_{Ke}^{f_{IS}} / K_{Ke}^{f_{U}}|}{|K_{Ke}^{f_{IS}}|} \end{pmatrix}. \quad (11)$$

Здесь k – количество вариантов описания архитектуры создаваемой ИС, сформированных в результате выполнения метода синтеза вариантов описаний архитектуры создаваемой ИС.

Строки матрицы выигрыша Поставщика (10) определяют значения функции удовлетворения Поставщика от реализации совокупности ИТ-услуг j -го варианта описания архитектуры ИС. Столбцы матрицы выигрыша Поставщика (10) определяют значения функции удовлетворения Поставщика от реализации i -го функционального требования к ИС в различных вариантах описания архитектуры данной ИС. Строки матрицы выигрыша Потребителя (11) определяют значения функции удовлетворения Потребителя от реализации совокупности ИТ-услуг j -го варианта описания архитектуры ИС. Столбцы матрицы выигрыша Потребителя (11) определяют значения функции удовлетворения Потребителя от реализации i -го функционального требования к ИС в различных вариантах описания архитектуры данной ИС.

В случае, если все описания фреймов, интерфейсов и связей, характеризующие Про в i -м функциональном требовании к ИС и используемые для формирования представления $K_{ji}^{f_{IS}}$, будут взяты из библиотеки описаний ранее реализованных требований к ИТ-услугам ИС [20], выигрыш Поставщика от реализации i -го функционального требования в j -м варианте описания архитектуры ИС будет равен 1. В случае, если все описания фреймов, интерфейсов и связей, характери-

зующие ПрО в i -м функциональном требовании к ИС и используемые для формирования представления K_{ji}^{fs} , являются для Поставщика новыми, то выигрыш Поставщика от реализации i -го функционального требования в j -м варианте описания архитектуры ИС будет равен 0. В случае, если новой для Поставщика является только часть описаний этих фреймов, интерфейсов или связей, либо новые описания фреймов и интерфейсов базируются на присутствующих в библиотеке описаниях фреймов, интерфейсов или связей характеризующих абстрактные термины ПрО, выигрыш Поставщика от реализации i -го функционального требования в j -м варианте описания архитектуры ИС будет находиться в промежутке между 0 и 1.

В случае, если все описания фреймов, интерфейсов и связей, характеризующие ПрО в i -м функциональном требовании к ИС и используемые для формирования представления K_{ji}^{fs} , будут отражать только ПрО Потребителя, выигрыш Потребителя от реализации i -го функционального требования в j -м варианте описания архитектуры ИС будет равен 1. Если все описания фреймов, интерфейсов и связей, характеризующие ПрО в i -м функциональном требовании к ИС и используемые для формирования представления K_{ji}^{fs} , являются для Потребителя новыми, то выигрыш Потребителя от реализации i -го функционального требования в j -м варианте описания архитектуры ИС будет равен 0. Если же новой для Потребителя является только часть описаний этих фреймов, интерфейсов или связей, либо новые описания фреймов и интерфейсов абстрагируют сформулированные Потребителем фреймы, интерфейсы или связи, выигрыш Потребителя от реализации i -го функционального требования в j -м варианте описания архитектуры ИС будет находиться в промежутке между 0 и 1.

Результатом игры (9) будет являться описание рациональной архитектуры ИС как набора IT-услуг, сформированное в результате нахождения исхода данной игры.

Для нахождения исхода игры (9) предлагается использовать метод поиска равновесий Нэша для биматричной игры в чистых стратегиях. Биматричный характер игры (8) определяет ее выигрыш как результат выбора таких стратегий, уклоняясь от которых ни Поставщик, ни Потребитель не смогут увеличить значение своих функций выигрыша. Подобная ситуация называется ситуацией равновесия по Нэшу. Следует также отметить, что стратегии игры (9) описывают конкретные варианты описаний архитектуры ИС как набора IT-услуг. Поэтому Поставщик и Потребитель заинтересованы в поиске детерминированного решения игры (9). Кроме того, данный метод является одним из наиболее простых методов поиска исходов биматричных игр.

Для определения ситуации равновесия по Нэшу в ходе синтеза описания рациональной архитектуры ИС уточним описания стратегий игры (9). Как следует из описания целевых функций (7) и (8), в общем случае данные стратегии определяются множеством общестемных представлений функциональных требований к ИС на уровне знаний, формирующих варианты описания архитектуры создаваемой ИС. Следовательно, любую стратегию $X_j^{(Pr,U)} \in \{X\}^{(Pr,U)}$ можно представить как сеть фреймов, представляющую собой результат выполнения операции (2) над множеством общестемных представлений

представлений $\{K_{ji}^{fs}\}_{i=(c+1),\dots,e}^{(Pr,U)}$, скорректированных по результату выполнения метода синтеза вариантов описания архитектуры создаваемой ИС.

Тогда ситуация равновесия по Нэшу в игре (9) будет представлять собой один или несколько вариантов описания архитектуры создаваемой ИС $\{K_{ji}^{fs}\}_{i=(c+1),\dots,e}^{(Pr,U)}$, для которых выполняется условие

$$\{K_{ji}^{fs}\}_i^{(Pr,U)} \in \underset{\{K_{ji}^{fs}\}_i^{(Pr,U)} \in \{K_{i1}^{fs}\}, \dots, \{K_{i1}^{fs}\}_i^{(Pr,U)}}{\text{arg max}} f^{(Pr,U)}(\{K_{-ji}^{fs}\}_i^{(Pr,U)} \parallel \{K_{ji}^{fs}\}_i^{(Pr,U)}), \quad (11)$$

где $j=1,\dots,k$, $i=(c+1),\dots,e$.

Исходя из данного условия, метод поиска равновесий Нэша в чистых стратегиях биматричной игры (9) будет состоять из следующих этапов.

Этап 1. Сформировать матрицу выигрышей Потребителя (11).

Этап 2. Сформировать матрицу выигрышей Поставщика (10).

Этап 3. Отметить максимальные элементы в каждом столбце матрицы (11). Если в некотором столбце данной матрицы несколько максимальных элементов, то необходимо отметить все такие элементы.

Этап 4. Отметить максимальные элементы в каждой строке матрицы (10). Если в некоторой строке данной матрицы несколько максимальных элементов, то необходимо отметить все такие элементы.

Этап 5. Провести поиск пересечения результатов Этапа 3 и Этапа 4 (то есть максимальных элементов, которые находятся на одних и тех же позициях в каждой из матриц).

Этап 6. Если результат выполнения Этапа 5 не является пустым множеством, зафиксировать соответствующие варианты описания архитектуры создаваемой ИС как описания рациональной архитектуры данной системы $\{K_{ji}^{fs}\}_{i=(c+1),\dots,e}^{(Pr,U)}$. В противном случае признать, что описания рациональной архитектуры ИС на чистых стратегиях не существует и завершить выполнение метода.

Этап 7. Если результатом выполнения Этапа 5 является набор из двух и более вариантов пересечения, то отсортировать их в порядке убывания вначале значений функции (8) для строк матрицы (11), содержащих эти варианты, а затем – значений функции (7) для строк матрицы (10), содержащих эти варианты. В противном случае – рассчитать значения функции (8) для строки матрицы (11), содержащей найденный вариант, и функции (7) для строки матрицы (10), содержащей найденный вариант. Завершить выполнение метода.

Ситуация, когда множество элементов, отмеченных в матрице (11) (максимальных по столбцам), и множество элементов, отмеченных в матрице (10) (максимальных по строкам) не пересекаются, означает, что равновесий в чистых стратегиях для данного варианта игры не существует. В этом случае предлагается рассматривать множество смешанных стратегий, поскольку равновесие Нэша существует для смешанного расширения любой конечной биматричной игры (9). В смешанном расширении биматричной игры всегда существует хотя бы одно равновесие по Нэшу. В ходе игры Поставщик и Потребитель должны использовать с положительными вероятностями только чистые стратегии, что позволяет свести поиск равновесий в

смешанных стратегиях к решению систем линейных уравнений и неравенств. Исходя из этого, для реализации Этапов 3–5 предложенного метода целесообразно использовать стандартные методы и алгоритмы, в частности – алгоритм Лемке-Хаусона [25].

7. Обсуждение результатов разработки модели и метода выбора варианта описания рациональной архитектуры создаваемой информационной системы

Использование рассмотренных в данном подразделе теоретико-игровых моделей, методов и алгоритмов поиска выигрышных чистых и смешанных стратегий должно обеспечить Поставщику и Потребителю в ходе игры выбор такого варианта описания архитектуры создаваемой ИС, который:

- а) соответствует выдвинутому Потребителем требованиям в максимальной степени;
- б) приносит Поставщику наибольший из возможных доходов от повторного использования ИТ-услуг и соответствующих ИТ-сервисов;
- в) обеспечивает реализацию непрерывных технологических цепочек сбора, хранения и обработки данных в рамках единой целостной ИС;
- г) гарантирует отсутствие ИТ-услуг и ИТ-сервисов, которые не являются обязательными в технологических цепочках сбора, хранения и обработки данных и не соответствует выдвинутым требованиям.

Недостатком применения метода поиска равновесий Нэша в чистых стратегиях биматричной игры (8) следует считать увеличение количества рисков принятия решения. Однако для рассмотренной задачи синтеза описания рациональной архитектуры ИС как совокупности ИТ-услуг данный недостаток не является критичным, поскольку процесс синтеза архитектуры предполагает координацию действий Поставщика и Потребителя после исхода игры с целью согласования и утверждения документов ИТ-проекта на создание ИС.

Таким образом, использование теоретико-игрового представления взаимодействия Поставщика и Потребителя позволяет формализовать процедуры формирования и корректировки такой архитектуры ИС как совокупности ИТ-услуг, который обеспечивает приемлемую для каждой стороны степень достижения глобальных целей. Данные результаты являются новыми и, как показано выше, не имеют аналогов в теоретических исследованиях и существующих системах управления требованиями.

8. Выводы

1. Предложено для решения задачи синтеза вариантов описания архитектуры создаваемой ИС использовать метод, основанный на дальнейшем развитии алгоритма решения задачи кластеризации CLOPE. Предлагаемый метод позволяет учитывать изначально неполноту множества сформулированных требований к ИС за счет отбора вариантов описаний, незначительно хуже оптимального. Это позволит выделить множество вариантов описания архитектуры создаваемой ИС, учитывающих частичное дублирование описаний отдельных требований. В результате такого выделения значительно расширяется область поиска компромиссного решения относительно исходного варианта описания архитектуры создаваемой ИС, основанного на множестве не связанных друг с другом требований к системе. Искомое компромиссное решение представляет собой вариант описания рациональной архитектуры создаваемой ИС, удовлетворяющий Поставщика и Потребителя в максимальной степени.

2. Разработаны формальные описания целей Поставщика (7) и Потребителя (8) создаваемой ИС. Эти описания позволяют рассматривать процесс проектирования архитектуры создаваемой ИС как процесс поиска такого варианта описания архитектуры, который обеспечит бы максимально возможные рассматриваемые значения функций (7) и (8) для множества требований к создаваемой ИС.

3. Разработана теоретико-игровая модель синтеза описания рациональной архитектуры создаваемой ИС (9). Эта модель позволяет осуществить процесс синтеза описания рациональной архитектуры как выбор варианта описания, в наибольшей степени соответствующего представлениям создаваемой системы Поставщиком и Потребителем. Использование данной модели позволяет автоматизировать процесс проектирования архитектуры системы путем поиска точек равновесия по Нэшу в чистых и смешанных стратегиях.

4. Разработан метод поиска точек равновесия по Нэшу в чистых стратегиях игры (9). Использование данного метода позволяет выбрать наиболее выгодный для Потребителя и Поставщика вариант описания рациональной архитектуры создаваемой ИС. Также рассмотрена возможная модификация разработанного метода для перехода от чистых к смешанным стратегиям игры (9).

Литература

1. ГОСТ ИСО/МЭК 15288–2005. Системная инженерия. Процессы жизненного цикла систем [Текст]. – Введ. 01–01–2007. – М.: Стандартинформ, 2006. – 57 с.
2. ГОСТ ИСО/МЭК 12207–2010. Системная и программная инженерия. Процессы жизненного цикла программных средств [Текст]. – Введ. 01–03–2012. – М.: Стандартинформ, 2011. – 106 с.
3. Руководство к своду знаний по управлению проектами (Руководство РМВОК): пятое издание [Текст]. – Newton Square: Project Management Institute, Inc., 2013. – 586 с.
4. COCOMO II Model Definition Manual [Electronic resource] // Center for Systems and Software Engineering. – Available at: http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CI_modelman2000.0.pdf – Last accessed: 15.01.2016. – Title from the screen.
5. Моделирование требований пользователей [Электронный ресурс] // Microsoft Developer Network. – Режим доступа: <https://msdn.microsoft.com/ru-ru/library/dd409376.aspx> – 15.01.2016. – Назв. с экран.

6. Vilpola, I. H. A method for improving ERP implementation success by the principles and process of user-centred design [Text] / I. H. Vilpola // Enterprise Information Systems. – 2008. – Vol. 2, Issue 1. – P. 47–76. doi: 10.1080/17517570701793848
7. Sutcliffe, A. Scenario-based requirements analysis [Text] / A. Sutcliffe // Requirements Engineering. – 1998. – Vol. 3, Issue 1. – P. 48–65. doi: 10.1007/bf02802920
8. Mansilla, D. A Proposal of a Process Model for Requirements Elicitation in Information Mining Projects [Text] / D. Mansilla, M. Pollo-Cattaneo, P. Britos, R García-Martínez // Lecture Notes in Business Information Processing. – 2013. – P. 165–173. doi: 10.1007/978-3-642-36611-6_13
9. Липко, Ю. Алгоритм формализации требований при разработке информационных систем [Текст] / Ю. Липко // Известия Южного федерального университета. Технические науки. – 2014. – № 6 (155). – С. 153–158.
10. Тюрганов, А. Г. Особенности формализации предметной области корпоративных информационных систем [Текст] / А. Г. Тюрганов // Вестник Уфимского государственного авиационного технического университета. – 2007. – Т. 9, № 5. – С. 72–76.
11. Cleland-Huang, J. Mining Domain Knowledge [Requirements] [Text] / J. Cleland-Huang // IEEE Software. – 2015. – Vol. 32, Issue 3. – P. 16–19. doi: 10.1109/ms.2015.67
12. Yue, T. A systematic review of transformation approaches between user requirements and analysis models [Text] / T. Yue, L. C. Briand, Y. Labiche // Requirements Engineering. – 2010. – Vol. 16, Issue 2. – P. 75–99. doi: 10.1007/s00766-010-0111-y
13. Ralph, P. The illusion of requirements in software development [Text] / P. Ralph // Requirements Engineering. – 2013. – Vol. 18, Issue 3. – P. 293–296. doi: 10.1007/s00766-012-0161-4
14. Rational Requisite Pro [Electronic resource]. – IBM developerWorks. – Available at: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Wbcd69e09400c_4f72_9665_66f116225986/page/Rational%20RequisitePro – Last accessed: 15.01.2016. – Title from the screen.
15. IBM Rational DOORS Next Generation. An efficient requirements management tool for complex systems [Electronic resource]. – IBM. – Available at: http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=SP&infotype=PM&appname=SWGE_RA_IR_USEN&htmlfid=RAD14128USEN&attachment=RAD14128USEN.PDF – Last accessed: 15.01.2016. – Title from the screen.
16. Cradle Overview [Electronic resource]. – 3SL – Available at: <https://www.threesl.com/en/cradle/index.php> – Last accessed: 15.01.2016. – Title from the screen.
17. Система управления требованиями Devprom Requirements [Электронный ресурс]. – Сайт компании DEVPROM. – Режим доступа: <http://devprom.ru/features/Система-управления-требованиями-Devprom-Requirements> – Дата доступа: 15.01.2016. – Заг. с экрана.
18. Мадорская, Ю. М. Системы управления требованиями: что и зачем? [Электронный ресурс] / Ю. М. Мадорская // ReqCenter.pro. Согласованные знания для практического использования. – Режим доступа: <http://edu.reqcenter.pro/?p=2433> – Заг. с экрана.
19. Luckham, D. The Beginnings of IT Insight: Business Activity Monitoring [Electronic resource] / D. Luckham. – Real Time Intelligence & Complex Event Processing. – Available at: <http://complexevents.com/media/articles/cep-article-three.pdf> – Last accessed: 15.01.2016. – Title from the screen.
20. Евланов, М. В. Разработка методов формирования представления сформулированного требования к информационной системе на уровне знаний [Текст] / М. В. Евланов // Восточно-Европейский журнал передовых технологий. – 2015. – Т. 4, № 3 (76). – С. 4–11. doi: 10.15587/1729-4061.2015.47535
21. Левыкин, В. М. Паттерны проектирования требований к информационным системам: моделирование и применение [Текст]: монография / В. М. Левыкин, М. В. Евланов, М. А. Керносов. – Харьков: ООО «Компанія «Сміт», 2014. – 320 с.
22. Yang, Y. CLOPE: A fast and Effective Clustering Algorithm for Transaction Data [Text] / Y. Yang, H. Guan, J. You // Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. – New York: ACM, 2002. – P. 682–687.
23. Левыкин, В. М. Мета модель функциональной структуры информационной системы [Текст] / В. М. Левыкин, М. В. Евланов // Нові технології. Науковий вісник Кременчуцького університету економіки, інформаційних технологій і управління. – 2006. – № 1(11). – С. 67–72.
24. Евланов, М. В. Глобальные цели поставщика и потребителя ИТ-услуг [Текст] / М. В. Евланов, О. Е. Неумывакина, А. Ю. Карамышева // Восточно-европейский журнал передовых технологий. – 2012. – Т. 5, № 2 (59). – С. 12–17. – Режим доступа: <http://journals.urau.ru/eejet/article/view/4137/3900>
25. Матвеев, В. А. Конечные бескоалиционные игры и равновесия [Текст]: уч. пос. / В. А. Матвеев. – Псков: ПГПИ им. С. М. Кирова, 2005. – 176 с.